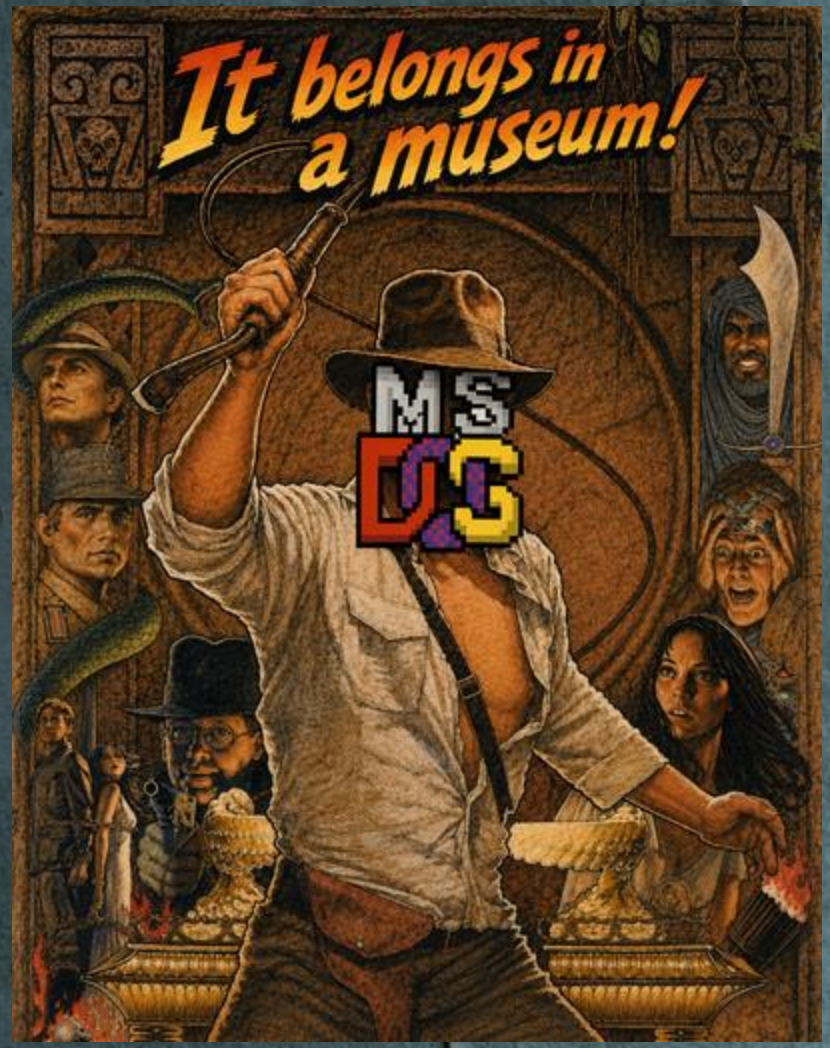
Software Archaeology – Two Tales of hacking 90s MS-DOS Software

Alon Livne & Matan Mansheroff



Who We Are

ALON LIVNE
CTO And Co-founder

MATAN MANSHEROFF
Senior Software Expert

Who We Do It For?

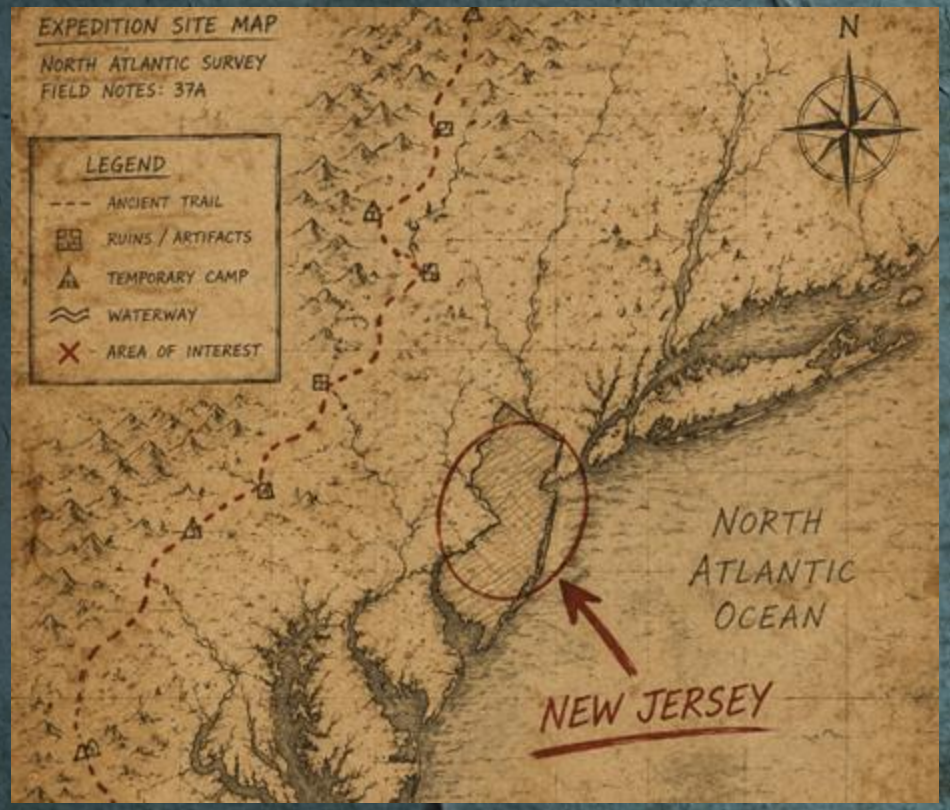
- Israeli start-up companies
- Microsoft
- Other such small companies

What We Do

- Deep tech consultation/development
- Vulnerability, firmware, embedded and other kinds of binary research
- Deep security reviews/audits
- Firmware and embedded research,
- CTFs

Agenda

- “Software Archeology”… ?
- Our “dig-sites” – *Kibbutz Dvir* and *New Jersey*
- Reverse engineering mid 90’s software
- Who cares anyway?
- Case studies – “QText” and “SAGA”



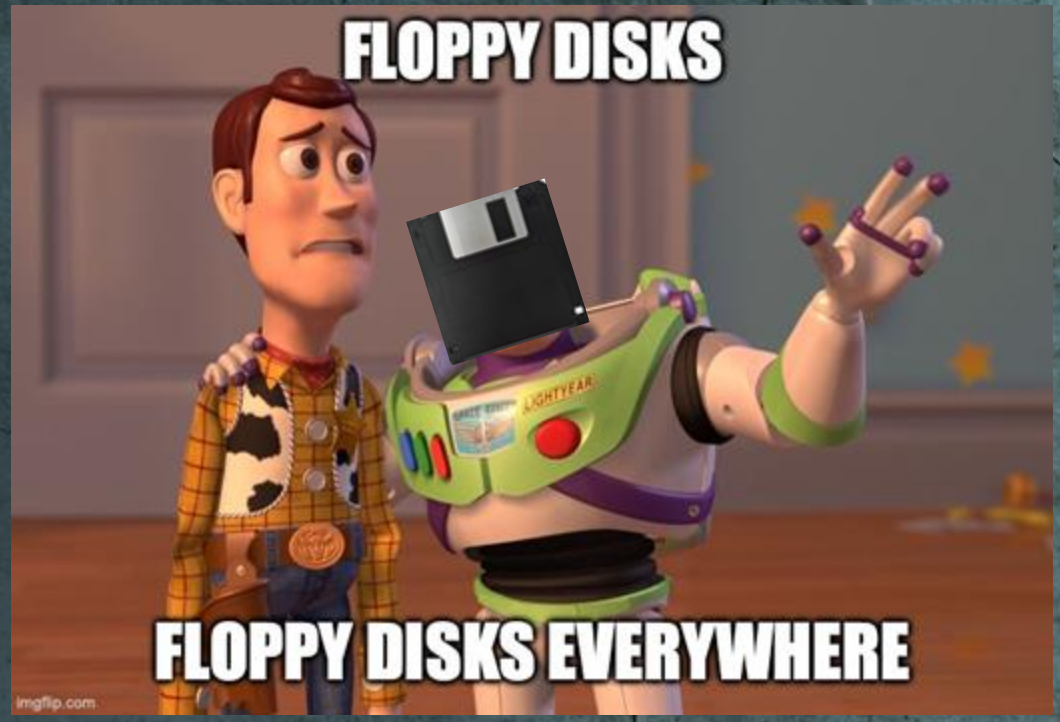
Motivations and Focus

What is to be gained, seen or learned?

- Our thought process
- Unpackers, obfuscators, emulators and old software
- Methodology
  - Reverse Engineering
  - Debugging and tracing
  - Web archeology
- Preservation of knowledge
- Have fun!

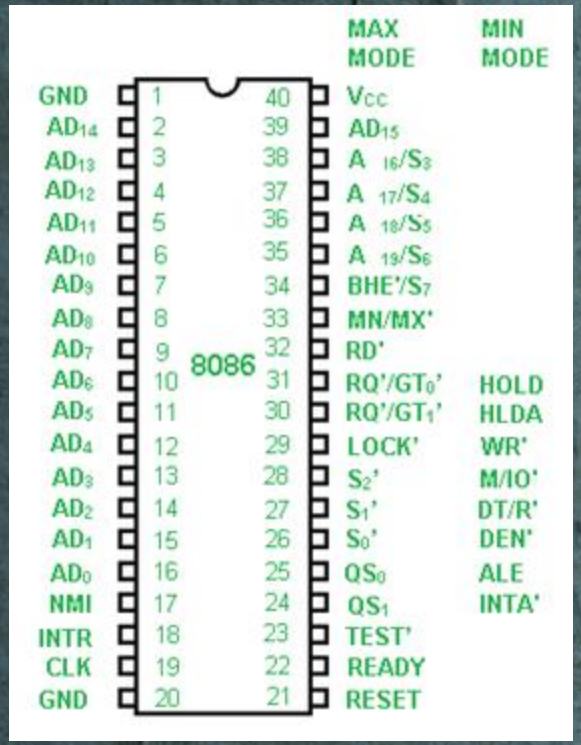
Getting our bearings

- It's the early 90s
- GUI is emerging
- ARPANET is still a thing
- BBSs are all the rage
- At the time – 8086 is king
- Our playing-field: MS-DOS



8086 - Overview

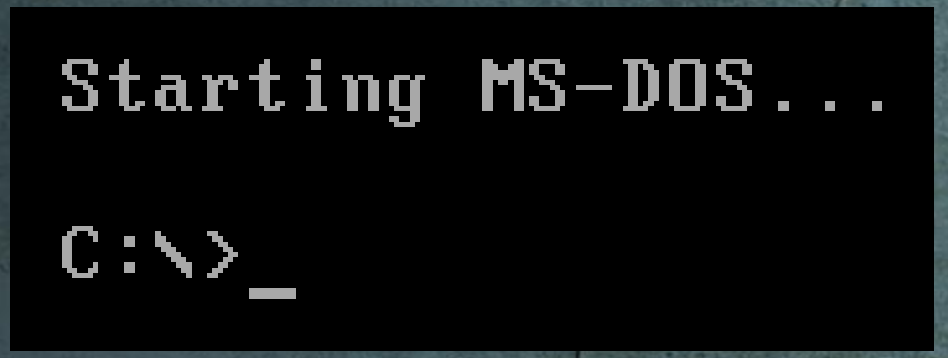
- First released in 1978 by Intel
- Originally just real mode
- 16 bit CPU, 20 bit memory bus
- 256 interrupt vectors
- x86 register model and instruction-set




```asm
mov dx, 03F8h    ; COM1 serial port base
in  al, dx       ; read byte from port
out dx, al       ; write byte to port
```

MS-DOS - Overview

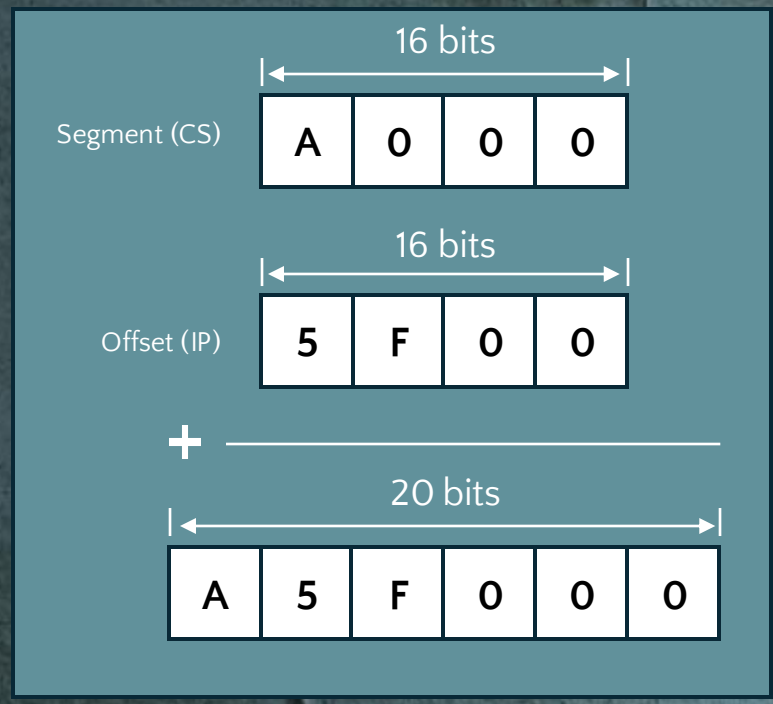
- “Disk operating system”
- Command-driven
- Some OS-ish functions
- Dead simple in all respects
- Business applications, games, etc



8086 – Real vs Protected Mode

Real Mode vs Protected Mode

Protected Mode (16/32-bit)	Real Mode (16-bit)	Feature
Flat + Segments (32-bit) 4 GB (32-bit)	Segment:Offset (20-bit)1 MB max	Addressing
Up to 4 GB	640KB app (≈ 1 MB)	Memory Limit
Rings 0–3 (user/kernel)	None (Ring 0 only)	Privilege Levels (Rings)
✓ MMU + Paging (virtual memory)	No MMU	Virtual Memory (Paging)
Protected segments (+ descriptors)	Simple segments (no protection)	Segmentation
IDT + Privileges	BIOS/DOS (INT 21h)	Interrupts



Case #1 - QText

- Client (IL) approaches us
- A trove of password-protected documents, credentials to which have been lost
- Created using QText, the first Hebrew word processor – from Dvir Software

Our mission – ***Decipher the documents!***

Deep Dive – QText - Demo

Jumping in

- We have a sample of an encrypted document
- Keyspace – 4 characters, limited
- Encryption adds a single line, 0x1D header to the program

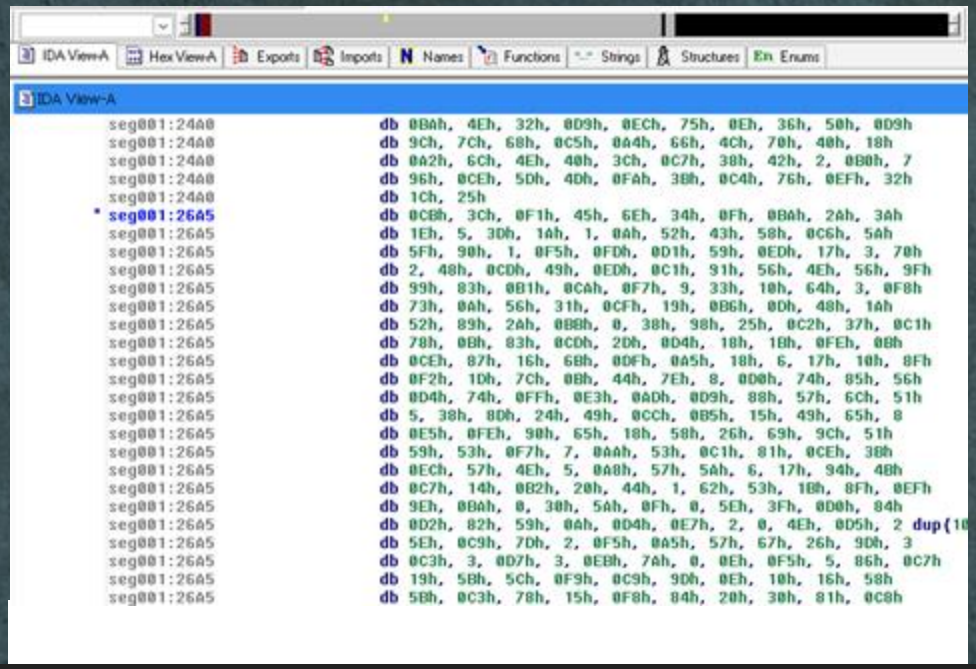
```
[➜  qtext  head -n1 ABC2.TXT | xxd
00000000: ff20 ff20 ff20 ff20 ff20 3030 2070 32e4  . . . . . 00 p2.
00000010: 2cb4 dc26 d0a0 a050 7080 4030 320d 0a    ,..&...Pp.@02..
[➜  qtext  head -n1 SPACE.TXT | xxd
00000000: ff20 ff20 ff20 ff20 ff20 3030 20db 5cc6  . . . . . 00 .\.
00000010: 3223 342c 6497 9062 28c1 3042 840d 0a    2#4,d..b(.0B...
```

- Brute-force should be easy
- How to execute a brute-force attack? And is there a better solution?

MS-DOS – static RE

- Is modern tooling good for this?
- Short answer – yes
- Long answer – NOOOOOOOOOO
- Fortunately – we are not alone in our use cases – ScummVM to the rescue with IDA Free 5.0

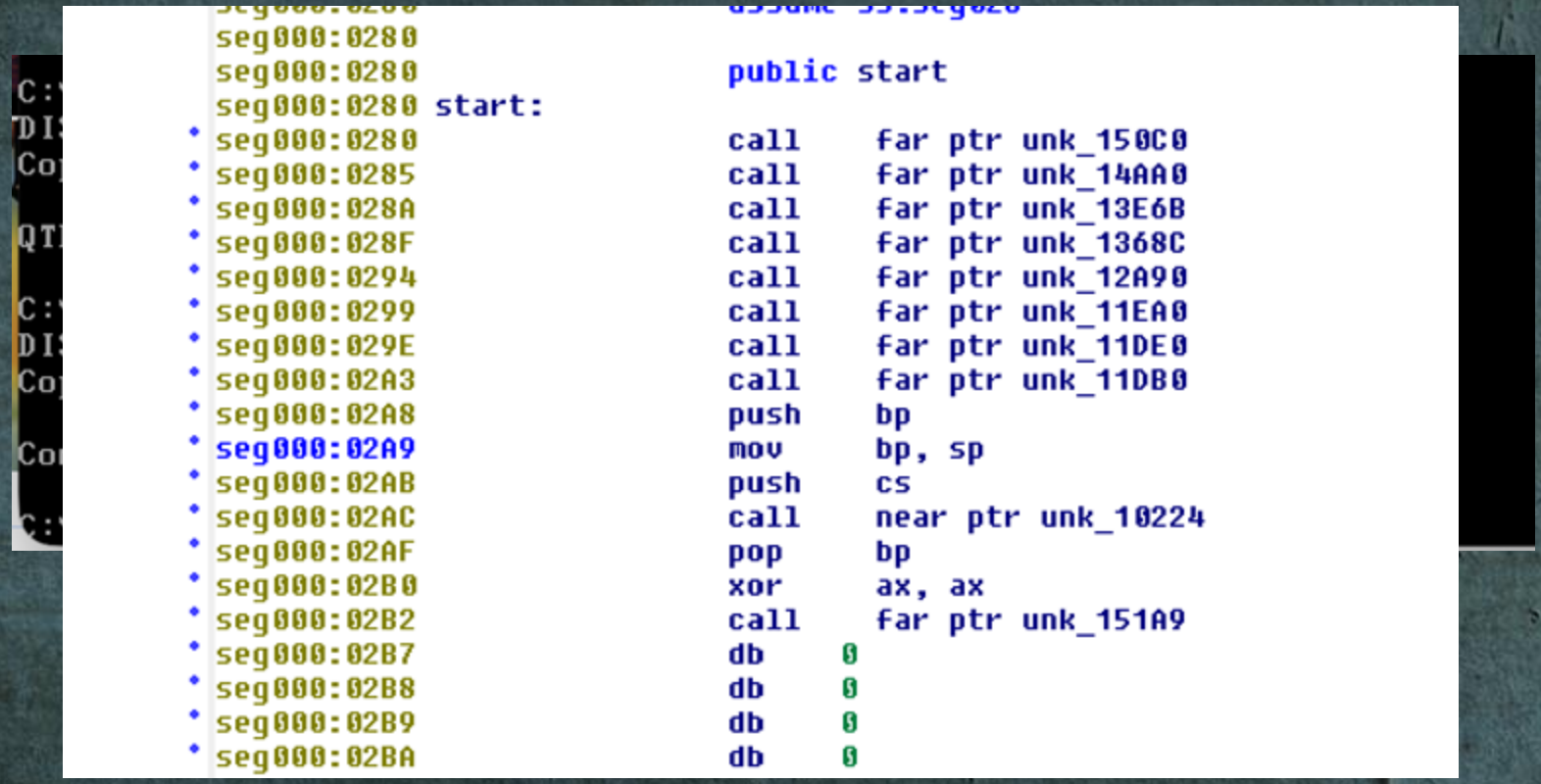
QText binary




```
^C%
➜  qtext file dos-qtext-5/QTEXT.EXE
dos-qtext-5/QTEXT.EXE: MS-DOS executable, MZ for MS-DOS Self-extracting PKZIP archive
➜  qtext
```

Web Archaeology

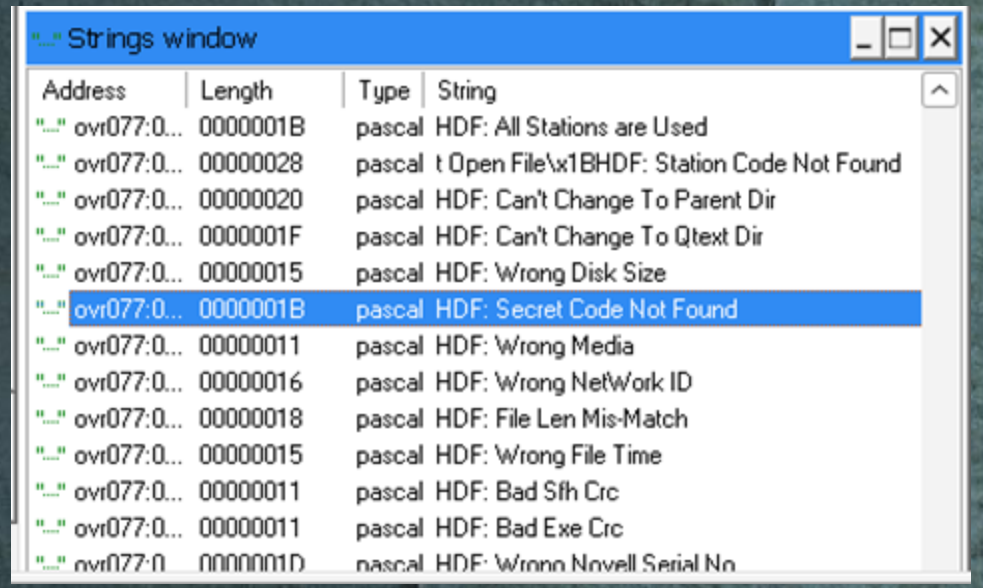
Unpacking QText



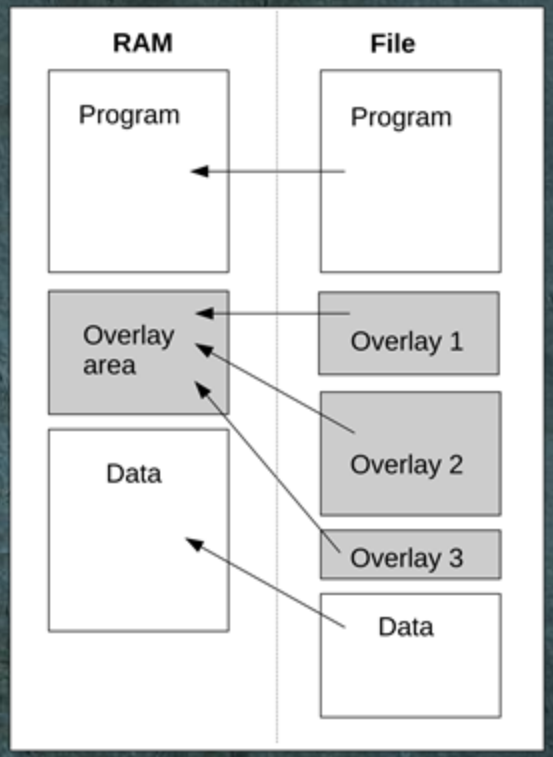
IDA goodies

tpdos	Turbo Pascal V5.0/5.5/6.0/7.0
tpdpmi	Turbo Pascal V7.0

```
mov     di, offset aPressAnyKey___ ; "\r\n\r\nPress ANY Key ... "
push    cs
push    di
xor     ax, ax          ; Logical Exclusive OR
push    ax
call    Write(Text &,String &,Word) ; Call Procedure
call    Write(Text &)   ; Call Procedure
call    READKEY(void)   ; Call Procedure
mov     [bp-6], al
```



Turbo Pascal Overlays



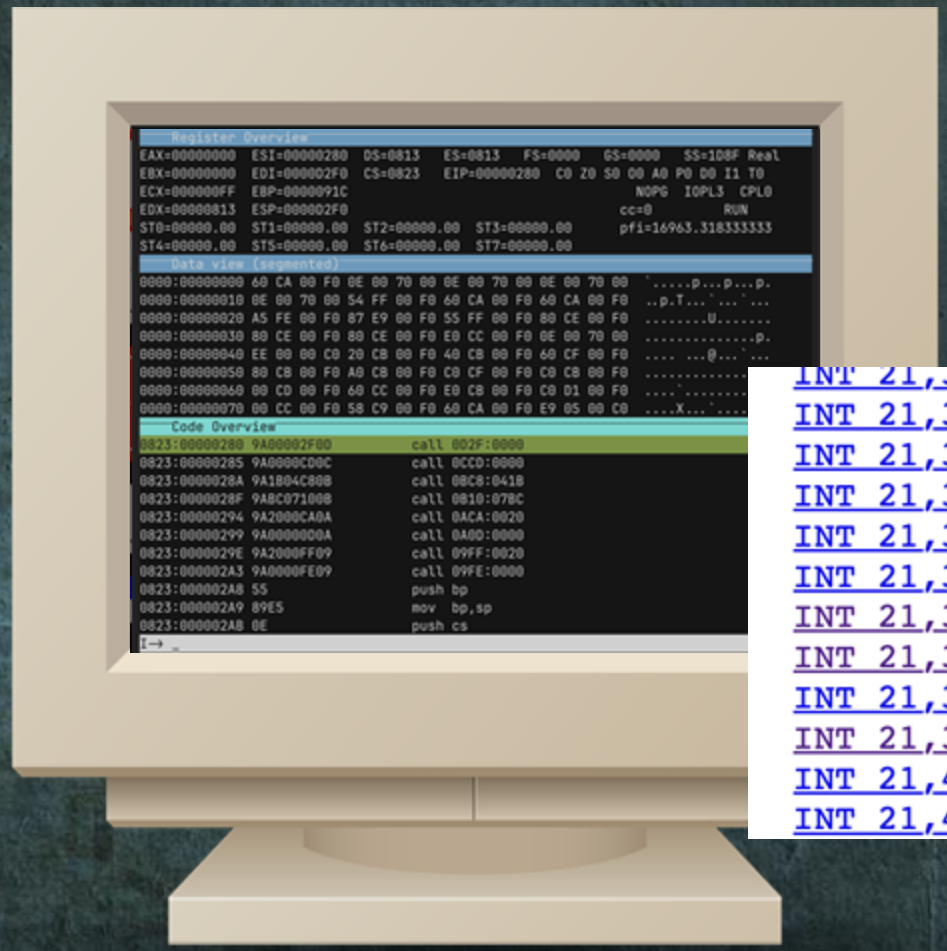

```
stub001:0000 ; ---------------------------------------
stub001:0000
stub001:0000 ; Segment type: Pure code
stub001:0000 stub001         segment byte public 'CODE'
stub001:0000                 assume cs:stub001
stub001:0000                 assume es:nothing, ss:noth
stub001:0000 stru_102C0      db 0CDh, 3Fh            ;
stub001:0000                 dw 0                    ;
stub001:0000                 dd 8                    ;
stub001:0000                 dw 696h                 ;
stub001:0000                 dw 64h                  ;
stub001:0000                 dw 1                    ;
stub001:0000                 dw 0                    ;
stub001:0000                 db 10h dup(0)           ;
stub001:0020
stub001:0020 ; ||||||||||||||| S U B R O U T I N E |||||
stub001:0020
stub001:0020 ; Attributes: thunk
stub001:0020
stub001:0020 sub_102E0       proc near               ;
stub001:0020                 jmp     far ptr sub_32F35
stub001:0020 sub_102E0       endp
stub001:0020
```

DOSBOX Debugger

```
Register Overview
EAX=00000000  ESI=00000280  DS=0813   ES=0813   FS=0000   GS=0000   SS=1D8F Real
EBX=00000000  EDI=0000D2F0  CS=0823   EIP=00000280  C0 Z0 S0 O0 A0 P0 D0 I1 T0
ECX=000000FF  EBP=0000091C                          NOPG  IOPL3  CPL0
EDX=00000813  ESP=0000D2F0                        cc=0         RUN
ST0=00000.00  ST1=00000.00  ST2=00000.00  ST3=00000.00      pfi=16963.318333333
ST4=00000.00  ST5=00000.00  ST6=00000.00  ST7=00000.00
Data view (segmented)
0000:00000000 60 CA 00 F0 0E 00 70 00 0E 00 70 00 0E 00 70 00  `.....p...p...p.
0000:00000010 0E 00 70 00 54 FF 00 F0 60 CA 00 F0 60 CA 00 F0  ..p.T...`...`...
0000:00000020 A5 FE 00 F0 87 E9 00 F0 55 FF 00 F0 80 CE 00 F0  ........U.......
0000:00000030 80 CE 00 F0 80 CE 00 F0 E0 CC 00 F0 0E 00 70 00  ..............p.
0000:00000040 EE 00 00 C0 20 CB 00 F0 40 CB 00 F0 60 CF 00 F0  .... ...@...`...
0000:00000050 80 CB 00 F0 A0 CB 00 F0 C0 CF 00 F0 C0 CB 00 F0  ................
0000:00000060 00 CD 00 F0 60 CC 00 F0 E0 CB 00 F0 C0 D1 00 F0  ....`...........
0000:00000070 00 CC 00 F0 58 C9 00 F0 60 CA 00 F0 E9 05 00 C0  ....X...`.....
Code Overview
0823:00000280 9A00002F0D          call 0D2F:0000
0823:00000285 9A0000CD0C          call 0CCD:0000
0823:0000028A 9A1B04C80B          call 0BC8:041B
0823:0000028F 9ABC07100B          call 0B10:07BC
0823:00000294 9A2000CA0A          call 0ACA:0020
0823:00000299 9A00000D0A          call 0A0D:0000
0823:0000029E 9A2000FF09          call 09FF:0020
0823:000002A3 9A0000FE09          call 09FE:0000
0823:000002A8 55                  push bp
0823:000002A9 89E5                mov  bp,sp
0823:000002AB 0E                  push cs
I→ _
```

```
seg000:0280                     assume ss:seg145
seg000:0280
seg000:0280                     public start
seg000:0280 start:                                      ; __SystemInit(void)
seg000:0280                     call    @__SystemInit$qv
seg000:0285                     call    @__CRTInit$qv   ; __CRTInit(void)
seg000:028A                     call    sub_13E6B
seg000:028F                     call    sub_1368C
seg000:0294                     call    sub_12A90
seg000:0299                     call    sub_11EA0
seg000:029E                     call    sub_11DE0
seg000:02A3                     call    sub_11DB0
seg000:02A8                     push    bp
seg000:02A9                     mov     bp, sp
seg000:02AB                     push    cs
seg000:02AC                     call    near ptr sub_10224
seg000:02AF                     pop     bp
seg000:02B0                     xor     ax, ax
seg000:02B2                     call    @Halt$q4Word    ; Halt(Word)
```

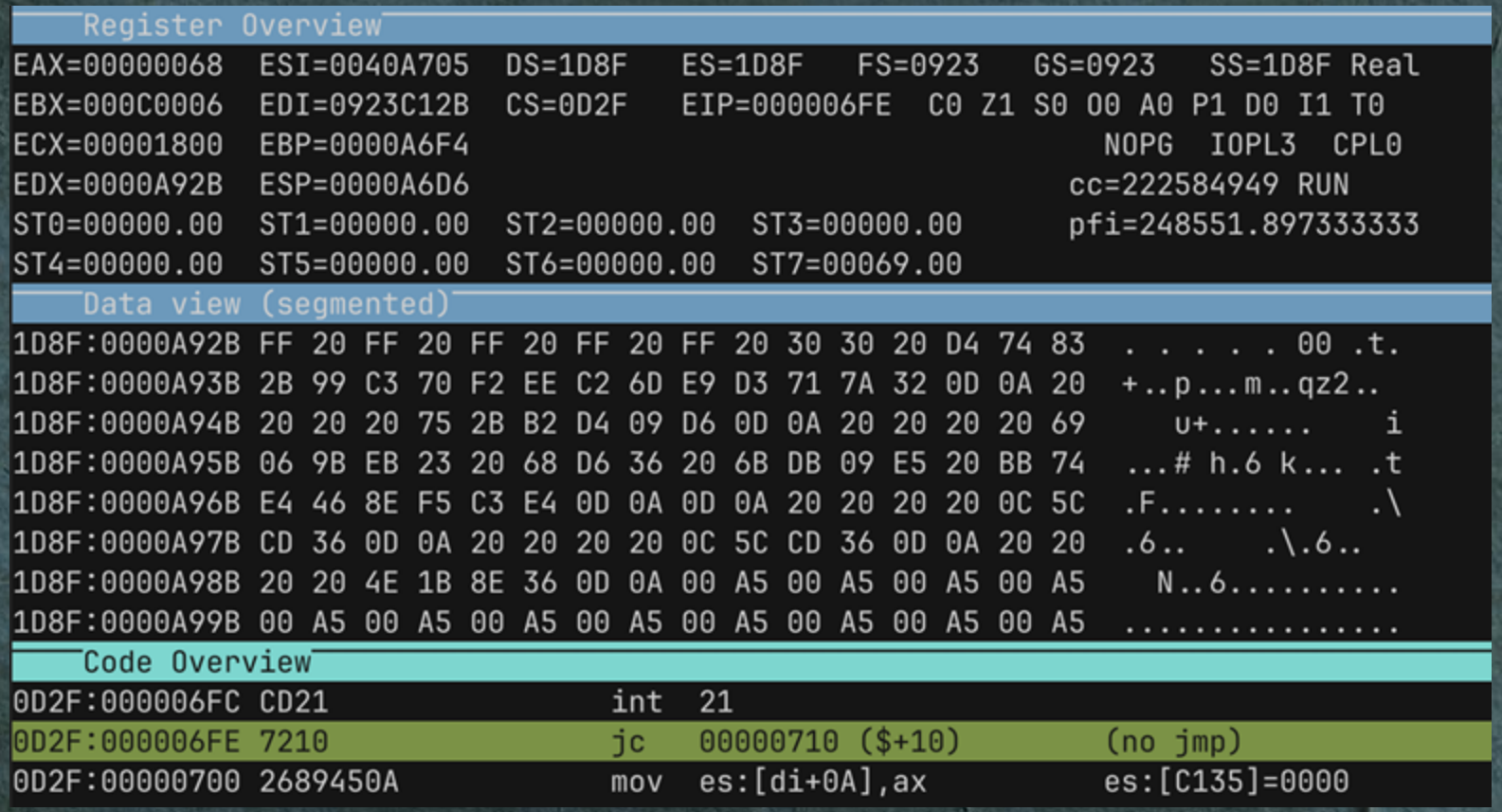
DOSBOX Debugger – breakpoints



1. Break on `int 21`
2. Break on `int 3F`
3. Break on root code

INT 21,36	Get disk free space
INT 21,37	Get/set switch character (undocumented)
INT 21,38	Get/set country dependent information
INT 21,39	Create subdirectory (mkdir)
INT 21,3A	Remove subdirectory (rmdir)
INT 21,3B	Change current subdirectory (chdir)
INT 21,3C	Create file using handle
INT 21,3D	Open file using handle
INT 21,3E	Close file using handle
INT 21,3F	Read file or device using handle
INT 21,40	Write file or device using handle
INT 21,41	Delete file

QText tracing



QText tracing

```
ovr055:05BB    call    Read(Text &,String &,Word)
ovr055:05C0    call    ReadLn(Text &)  ; Call Pro
ovr055:05C5    call    sub_138A3       ; Call Pro
ovr055:05CA    lea     di, [bp+var_151] ; Load Ef
ovr055:05CE    push    ss
ovr055:05CF    push    di
ovr055:05D0    les     di, [bp+arg_C]  ; Load Ful
ovr055:05D3    push    es
ovr055:05D4    push    di
ovr055:05D5    les     di, [bp+arg_8]  ; Load Ful
ovr055:05D8    push    es
ovr055:05D9    push    di
ovr055:05DA    les     di, [bp+arg_4]  ; Load Ful
ovr055:05DD    push    es
ovr055:05DE    push    di
ovr055:05DF    les     di, [bp+arg_0]  ; Load Ful
ovr055:05E2    push    es
ovr055:05E3    push    di
ovr055:05E4    push    cs
ovr055:05E5    call    near ptr decrypt_document
ovr055:05E8    cmp     [bp+arg_1C], 0  ; Compare
```

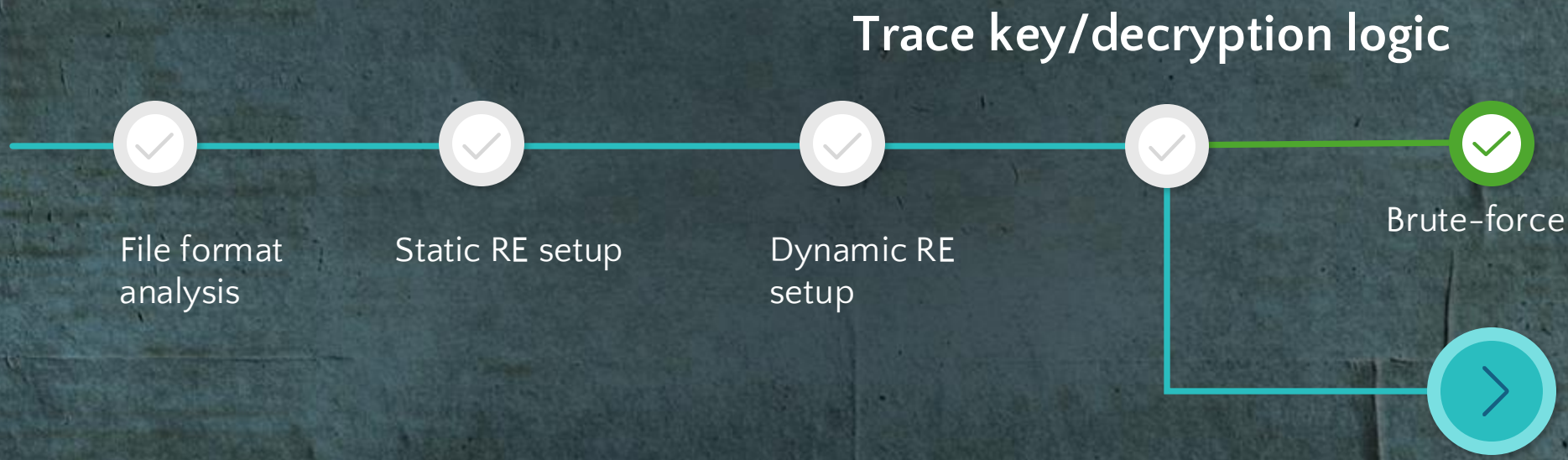
Key derivation

1. Mix passcode bytes
2. Concat them 4 times
3. Mix again

- Mixing function – cross-sums characters
- Validity bitmap – “nudges” invalid characters

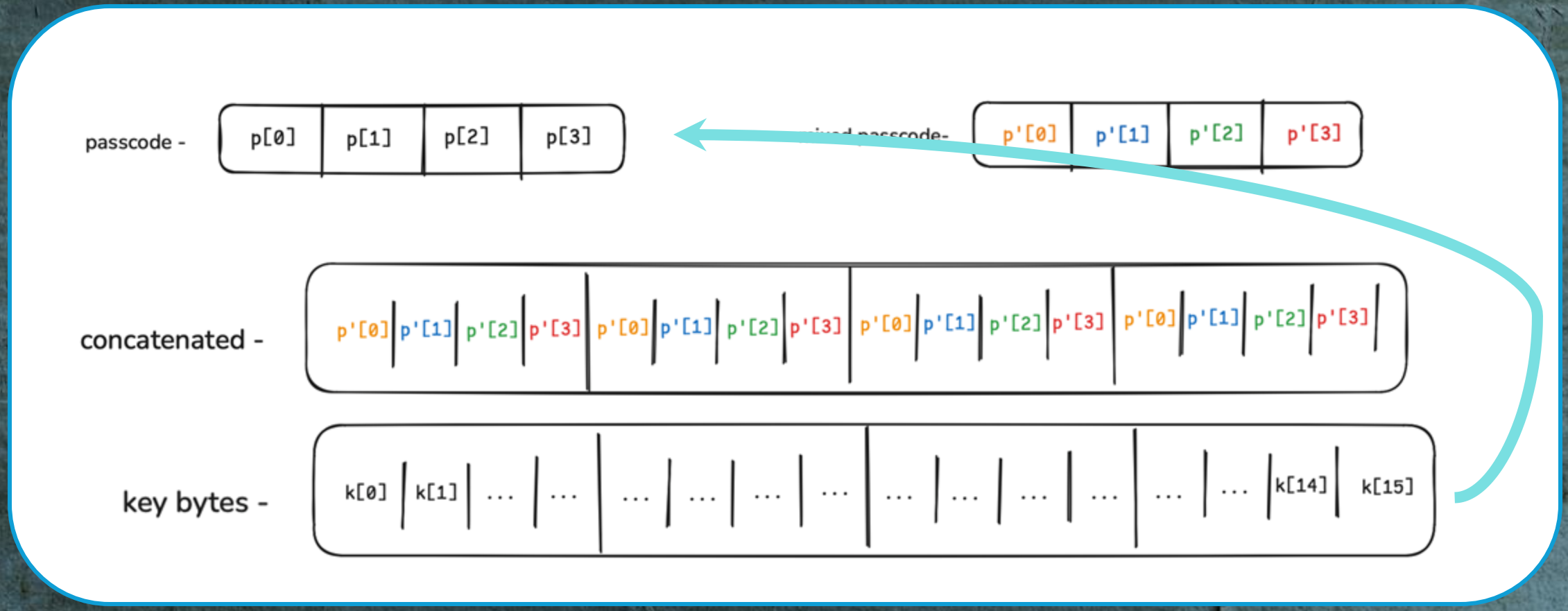
```python
def mix(input_passcode, modifier=0x22):
    passcode = bytearray(input_passcode)  # copy
    for i in range(len(passcode)):
        for j in range(len(passcode)):
            passcode[i] = (passcode[i] + passcode[j]) & 0xFF
        while not is_valid_char(passcode[i]):
            passcode[i] = (passcode[i] + modifier) & 0xFF
    return passcode


def expand(mixed_passcode):
    return mix(mixed_passcode * 4).hex()


def is_valid_char(c): ...  # 0x22 < c < 0x100
```

Change of plans!

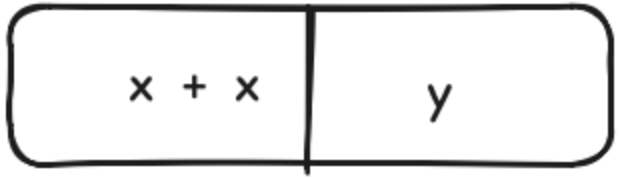


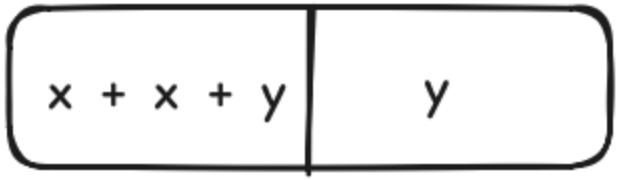
Cryptanalysis!

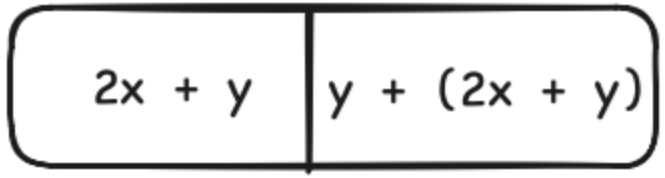
Cryptanalysis

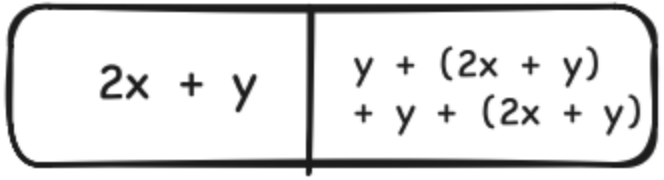


Cryptanalysis – mixing function

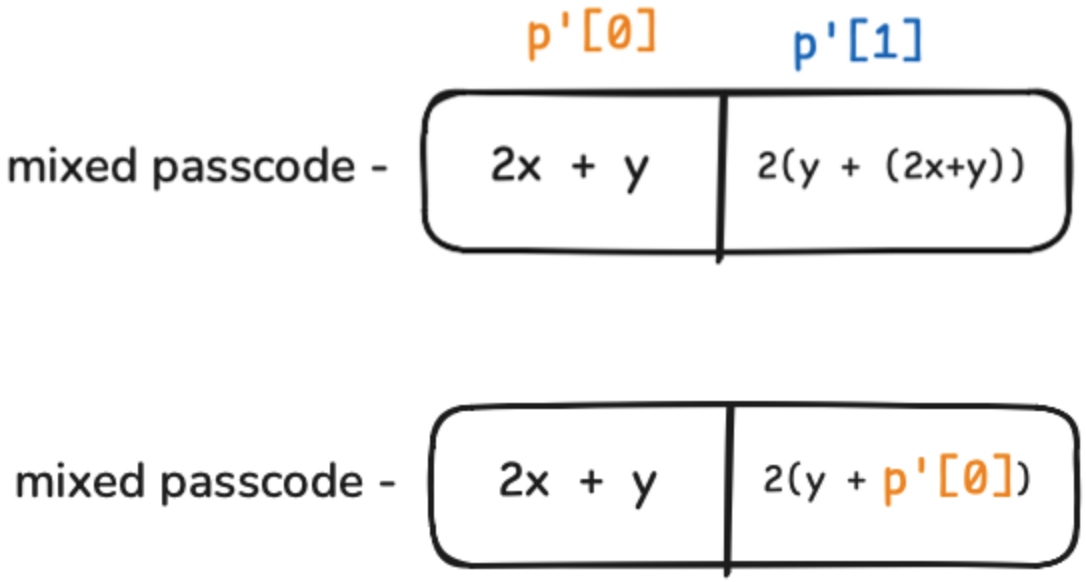
iteration 1 - 

iteration 2 - 

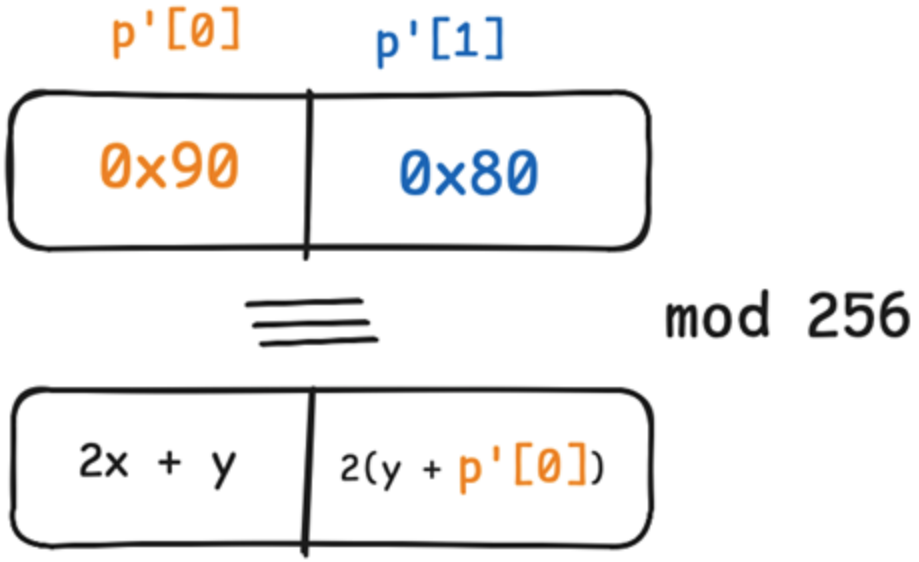
iteration 3 - 

iteration 4 - 

Cryptanalysis



Cryptanalysis



Cryptanalysis

```
In [1]: hex(0x40 * 2 % 256)
```

$y \equiv 0x40 - 0x90 \mod 256$

$y \equiv 0xc0 - 0x90 \mod 256$

$y = 0xb0 \text{ OR } 0x30$

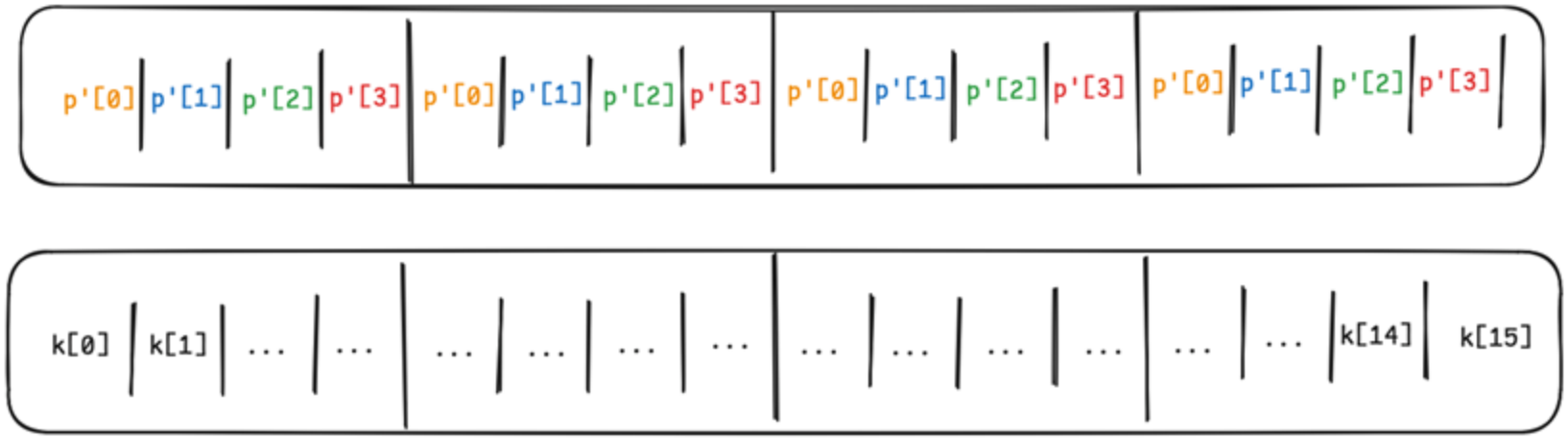
y

Cryptanalysis – generalized

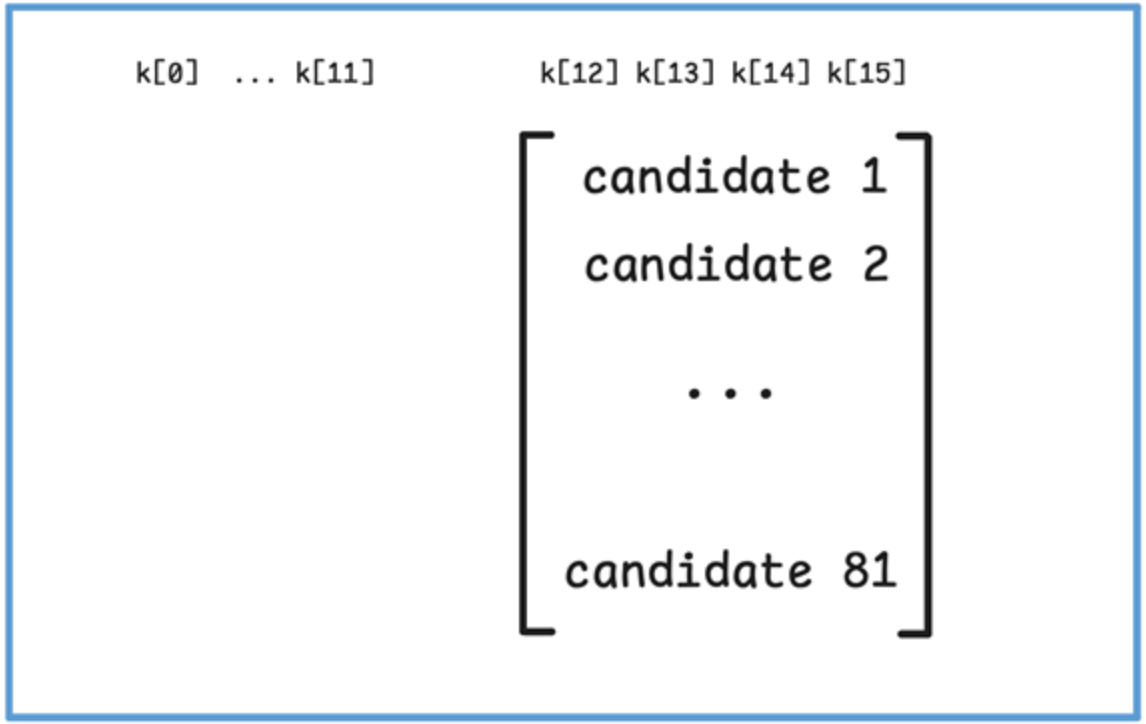
$$(1) \quad p[n] = p'[n]/2 - \sum_{0}^{n-1} p'[i] \mod 256$$

$$(2) \quad p[n] = (0x100 + p'[n]/2) - \sum_{0}^{n-1} p'[i] \mod 256$$

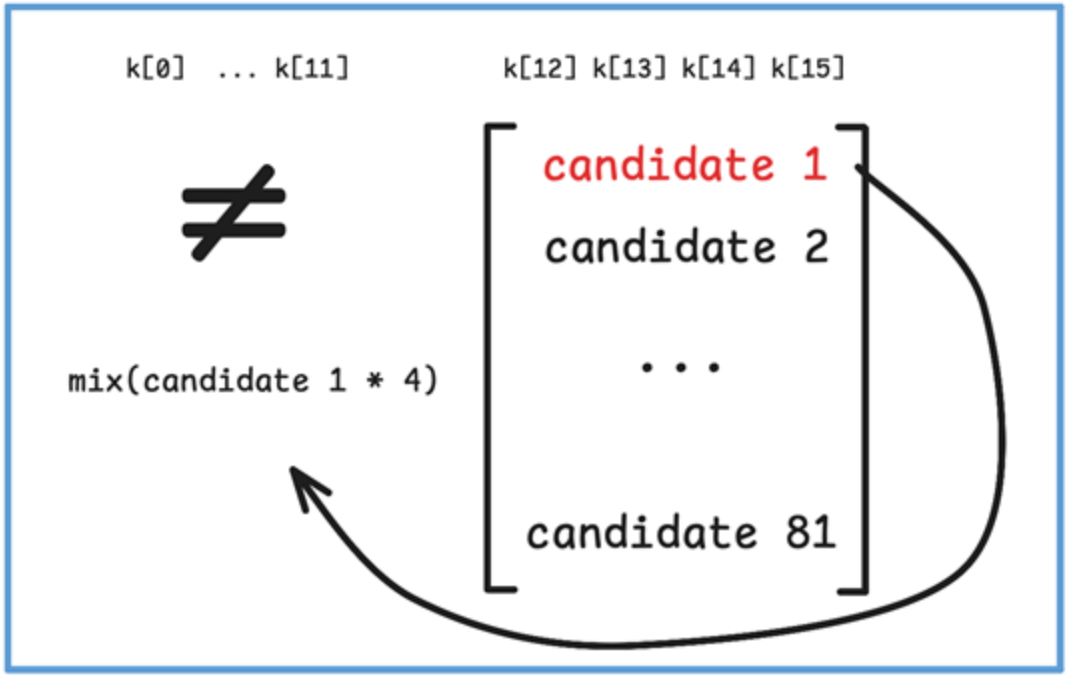
Cryptanalysis



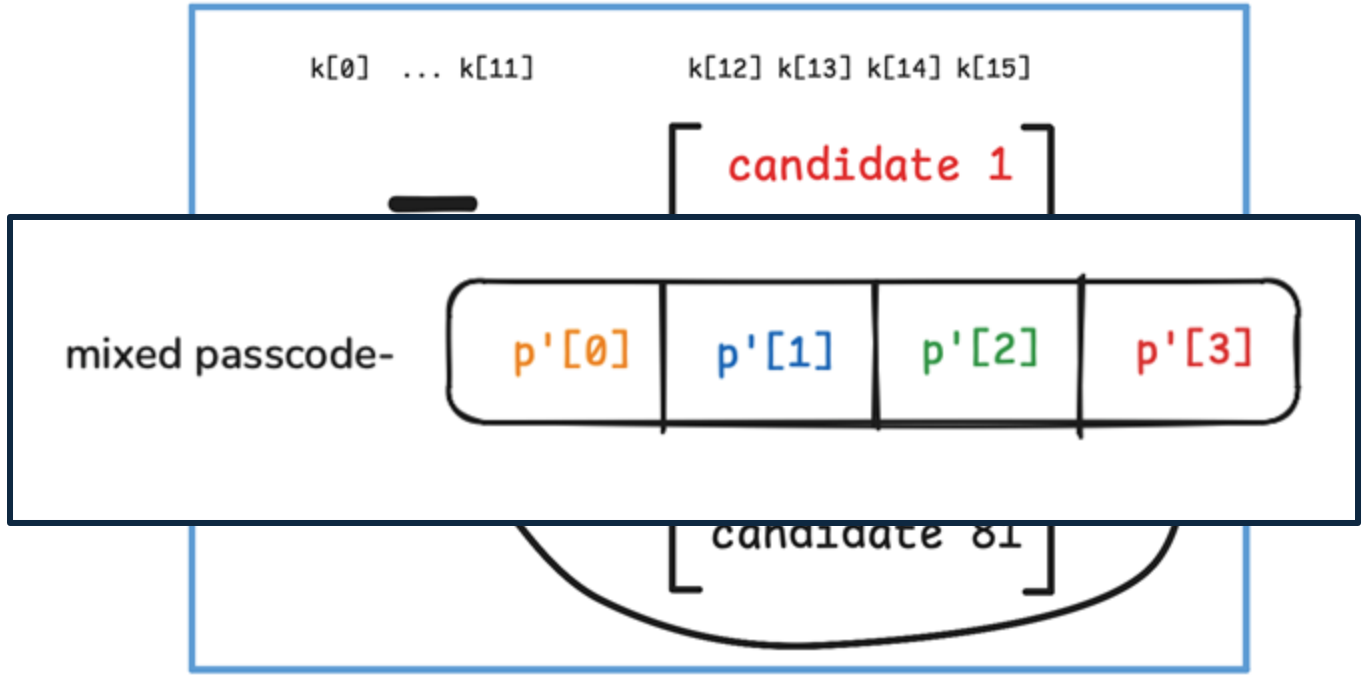
Cryptanalysis



Cryptanalysis



Cryptanalysis



Cryptanalysis done

$\sim 2^{13}$

Cryptanalysis

```
VALID_BYTES = b"ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890,."
```

$y \equiv 0x40 - 0x90 \mod 256$ $y \equiv 0xc0 - 0x90 \mod 256$

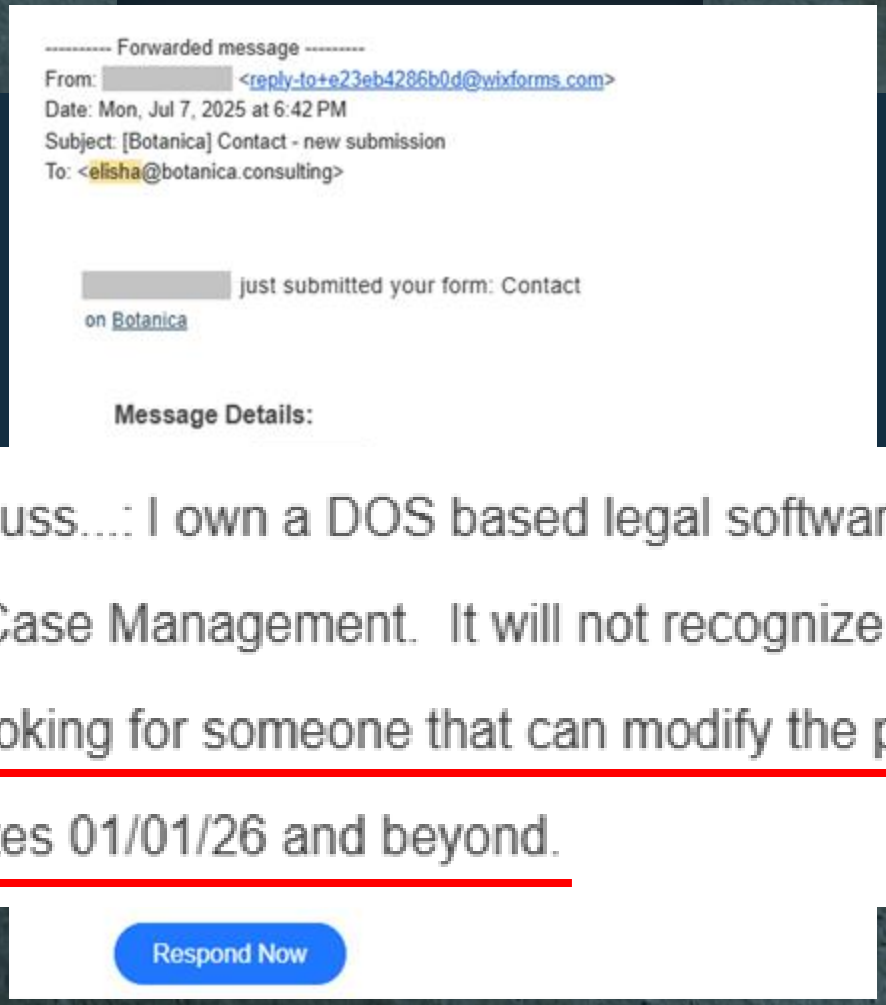
$y \equiv \ldots xb0$ OR $0x30$

Cryptanalysis - solved

$\sim 2^7$

Deep Dive – QText – Summary

SAGA – How It All Started



---------- Forwarded message ---------
From: [redacted] <reply-to+e23eb4286b0d@wixforms.com>
Date: Mon, Jul 7, 2025 at 6:42 PM
Subject: [Botanica] Contact - new submission
To: <elisha@botanica.consulting>

[redacted] just submitted your form: Contact
on Botanica

Message Details:

I'd like to discuss...: I own a DOS based legal software from 1999 called Saga Case Management. It will not recognize a date post 12/31/25. Looking for someone that can modify the program to recognize dates 01/01/26 and beyond.

Respond Now

Saga Case Management.

[–] h8br33der85 1 point 11 months ago
Lol awesome

[–] networkjack 1 point 11 months ago
If it works, why change?
permalink embed save report

Hacking_Tutorials
COMMENTS

[Removed
[removed]
2 comments

Saga Case Management
Need help reverse engineering this law firm case management software developed in 1999 using DOS. I own the software but it will not recognize a date past 12/21/25.

all 2 comments
sorted by: best

[–] Hacking_Tutorials-ModTeam [score hidden] 11 months ago - stickie
We are not your personal army. Next such violation and
banned permanently. Thank you!
permalink embed save report

[–] ColdPumpkin9679 1 point 11 months ago
I don't deal with doctors, lawyers or builders if I can. Not worth the hassle.

[–] [deleted] 3 points 11 months ago
Just say no to doctors and lawyers. It took us 20 years to figure that out.
permalink embed save report

[–] Flybinyte 1 point 11 months
Run away!!!!

[–] torind2000 1 point 11 months ago
Run away as fast as possible.
permalink embed save report

[–] MuthaPlucka 27 points 11 months ago
Run the other way.
permalink embed save report

[–] jeffa1792 12 points 11 months ago
Seriously
permalink embed save parent report

DOS?? (self.msp)
submitted 11 months ago by LynxGeekNYC
I got a potential law firm client that still uses DOS as a case management sy
Specifically SAGA Case Management system. I'm in shock. Who else runs i

[–] The-IT_MD 1 point 11 months ago
I ran into a DOS machine about 10 yrs ago; running on a 386 beige-box. It was used to control some scales that weighed oil-c
tubs of industrial paint.
This was for a £30m company.

[–] gmerideth 1 point 11 months ago
Yo.
Client using DOS6 app in a Windows XP VM (to hide from network scanner looking for outdated machines) to download (FTP) updated
financial data which produces some shitty output. Since he can't save to anything but a Borland Database he takes screen shots, sends
to his secretary who has to manually re-type all of the data.
They think it's perfectly normal.

There is a New Practice Manager for Personal Injury Firms in New York
Michael Zucchi
Published Oct 22, 2014
+ Follow
Good news for NY PI firms, there is now another choice to upgrade from SAGA.

[–] webjocky 12 points 11 months ago
Yikes.
Saga was acquired by Client Profiles, who were then acquired by Aderant Holdings, who then launched their own replacement called Expert Case, essentially they just bought their competition.
Good luck?
permalink embed save report

[–] floswamp 4 points 11 months ago
So, there's hope for a free upgrade?
permalink embed save parent report

Twitter.com
twitter.com › hashtag › pifirms
#pifirms - Twitter Search / Twitter
This Is Everything You Need to Know About the End of #SAGA Case Management.
http://ow.ly/FV6aq http://ow.ly/i/7WGp2 #pifirms #legal.

[–] sonicc_boom 1 point 11 months ago
But how....

[–] GetAfterItForever 8 points 11 months ago
Holy F—
permalink embed save report

[–] Al7amdulillaah 1 point 11 months ago
Is this a US thing?
permalink embed save parent report

[–] NovelRelationship830 1 point 11 months ago
Lawyers and Doctors here have a reputation for being very cheap with IT.
permalink embed save parent report

[–] First-Structure-2407 3 points 11 months ago
Bet it runs perfectly. Jealous.
permalink embed save report

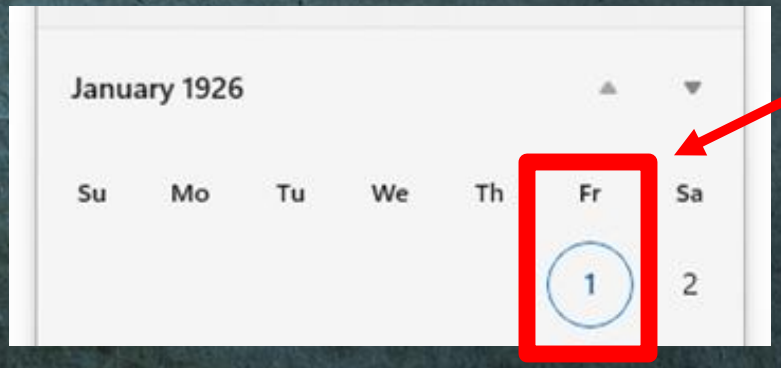
[–] VNJCinPA 2 points 11 months ago
Seriously... Super jealous. Back when Microsoft had programmers and not UI Designers
permalink embed save parent report

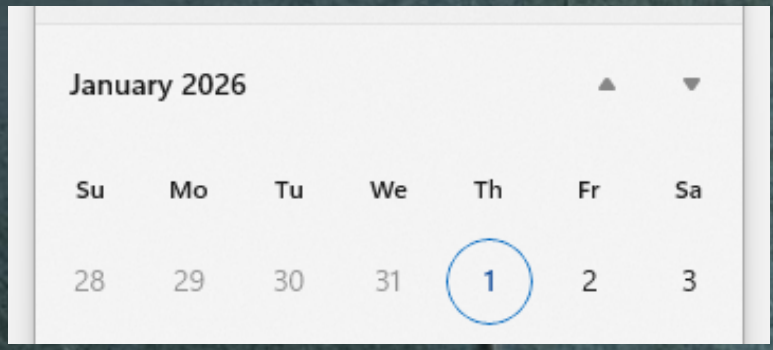
Saga Case Management.

Y2K Bug

Looking for someone that can modify the program to recognize dates 01/01/26 and beyond.

From Date: 01/01/26 Friday





Y2K Bug

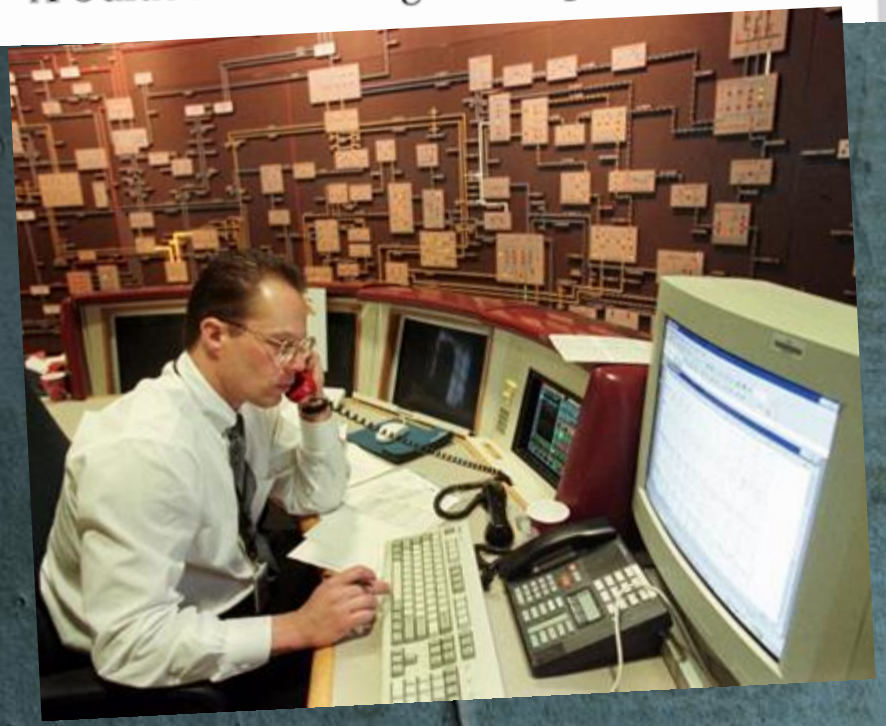
People prepared to "bug out"

A mobile home; a year's su[illegible] dehydrated food; a propane generator — those were just some of the precautionary purchases California computer programmer Scott Olmstead made in advance of 2000. (He also said he was shopping for a handgun.)

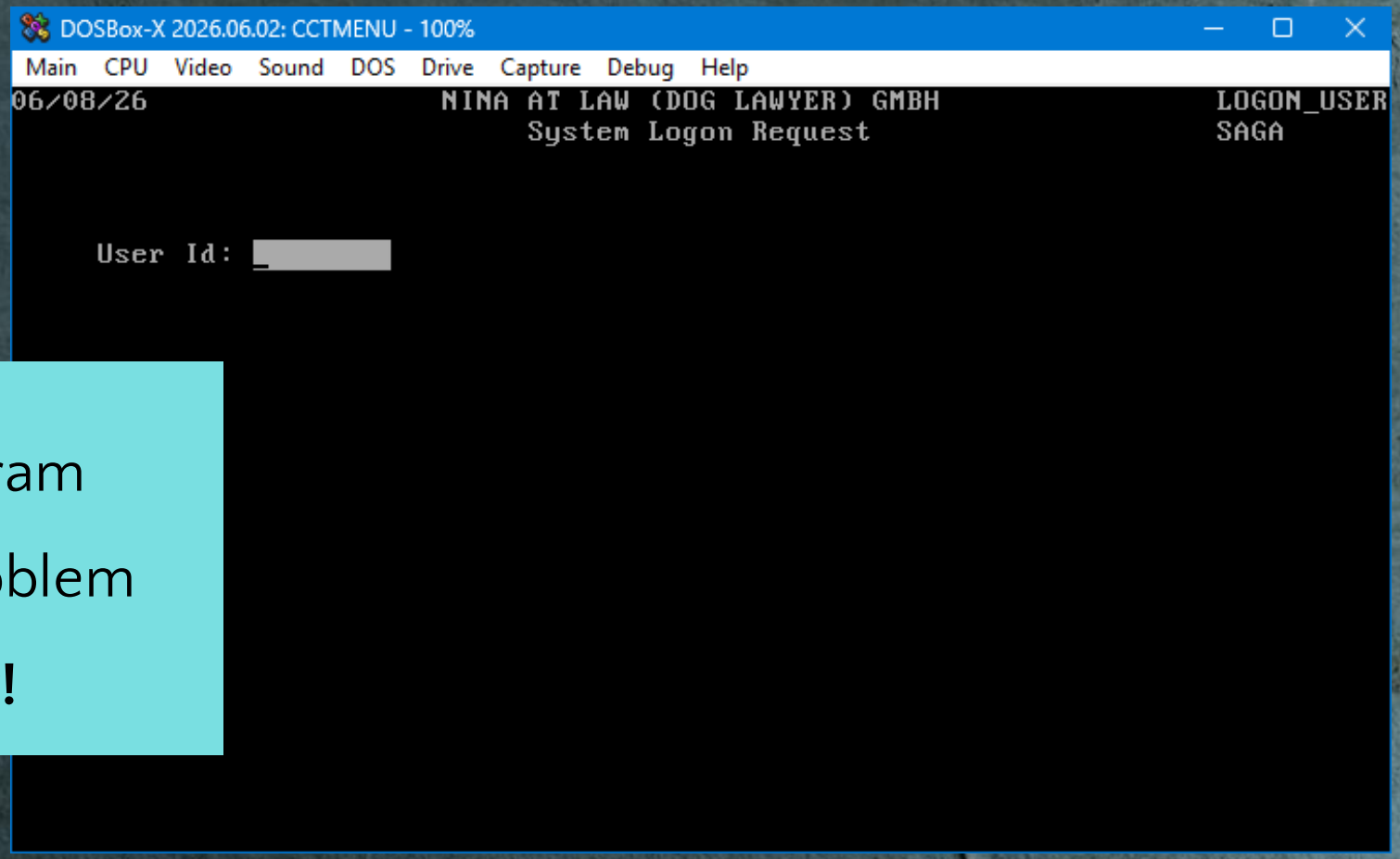
1998 February

IBM

The Year 2000 and 2-Digit Dates: A Guide for Planning and Implementation



SAGA – Kicking off



- Got to know the program
- Demonstrated the problem
- **Let's start researching!**

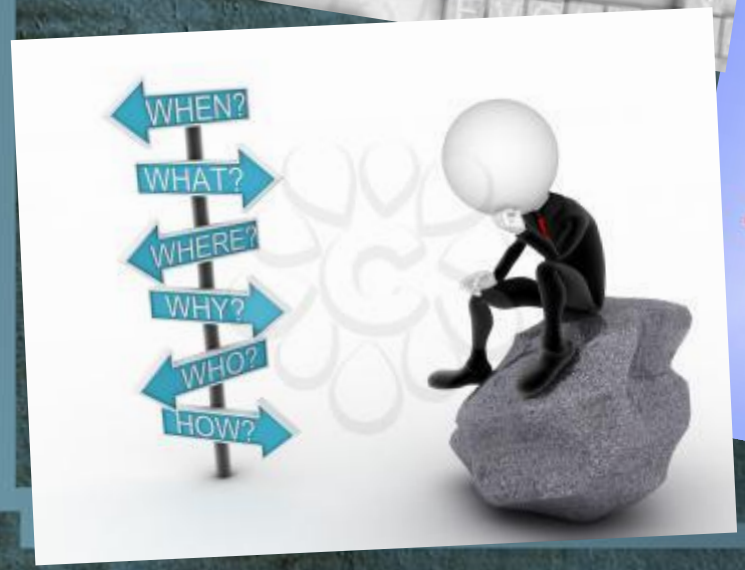

IDEAS
ACTION
CHANGE
+
=

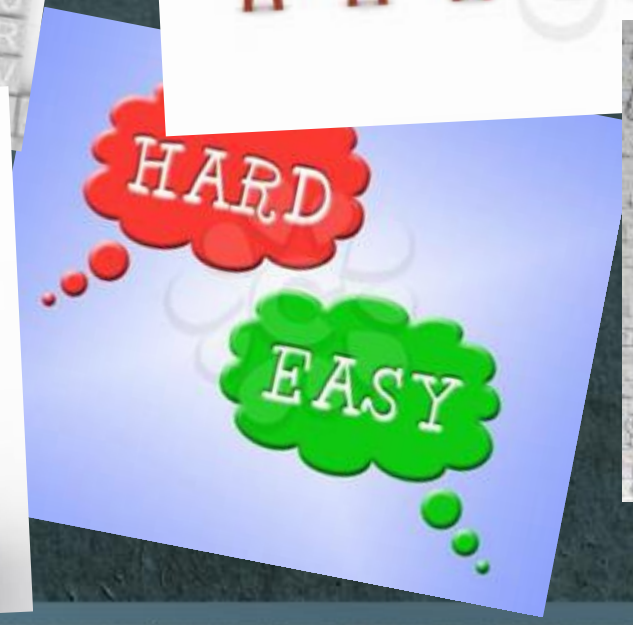

YES OR NO


LOGIC

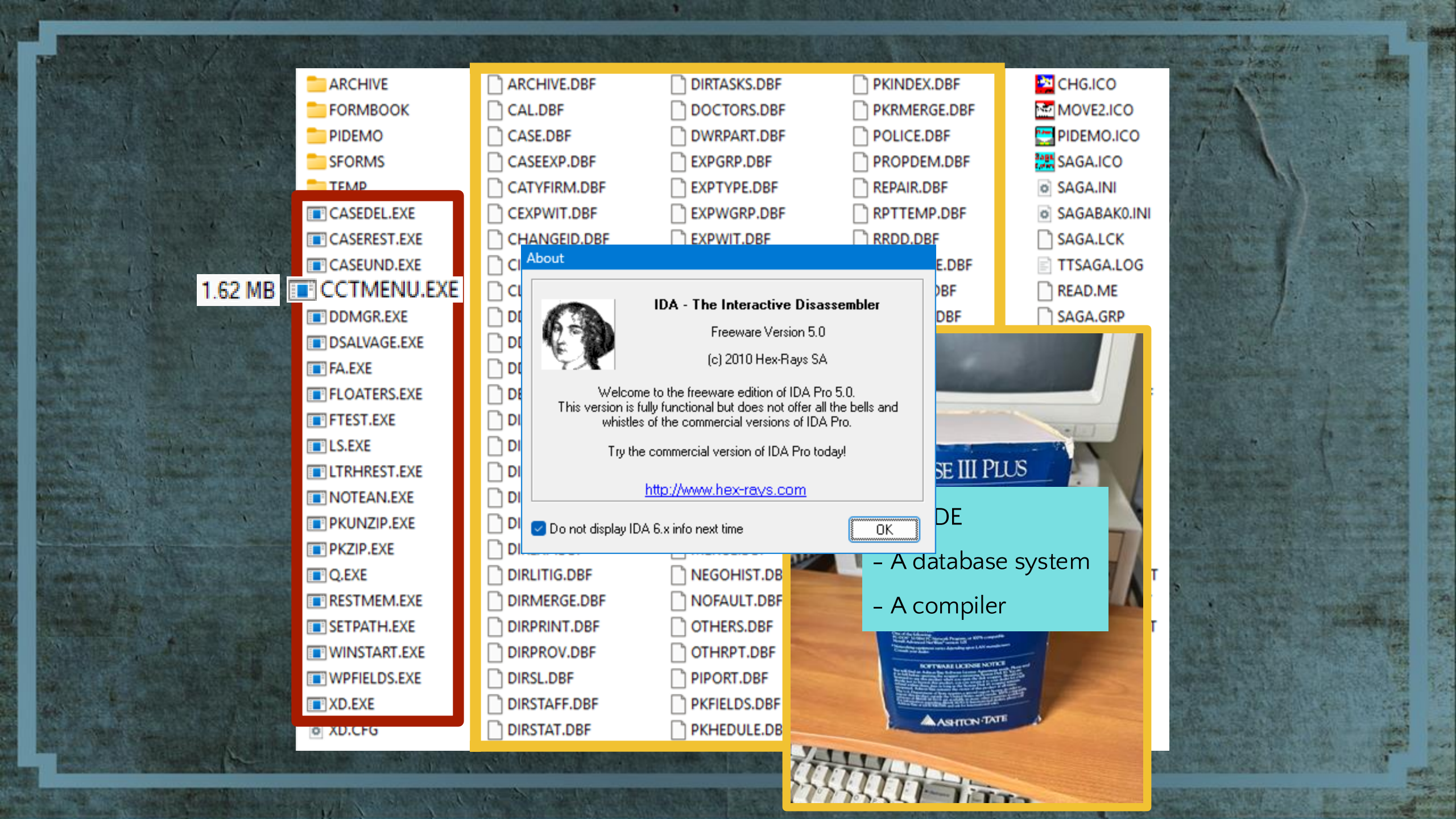

IMPROVE
ADAPTABILITY
FUTURE

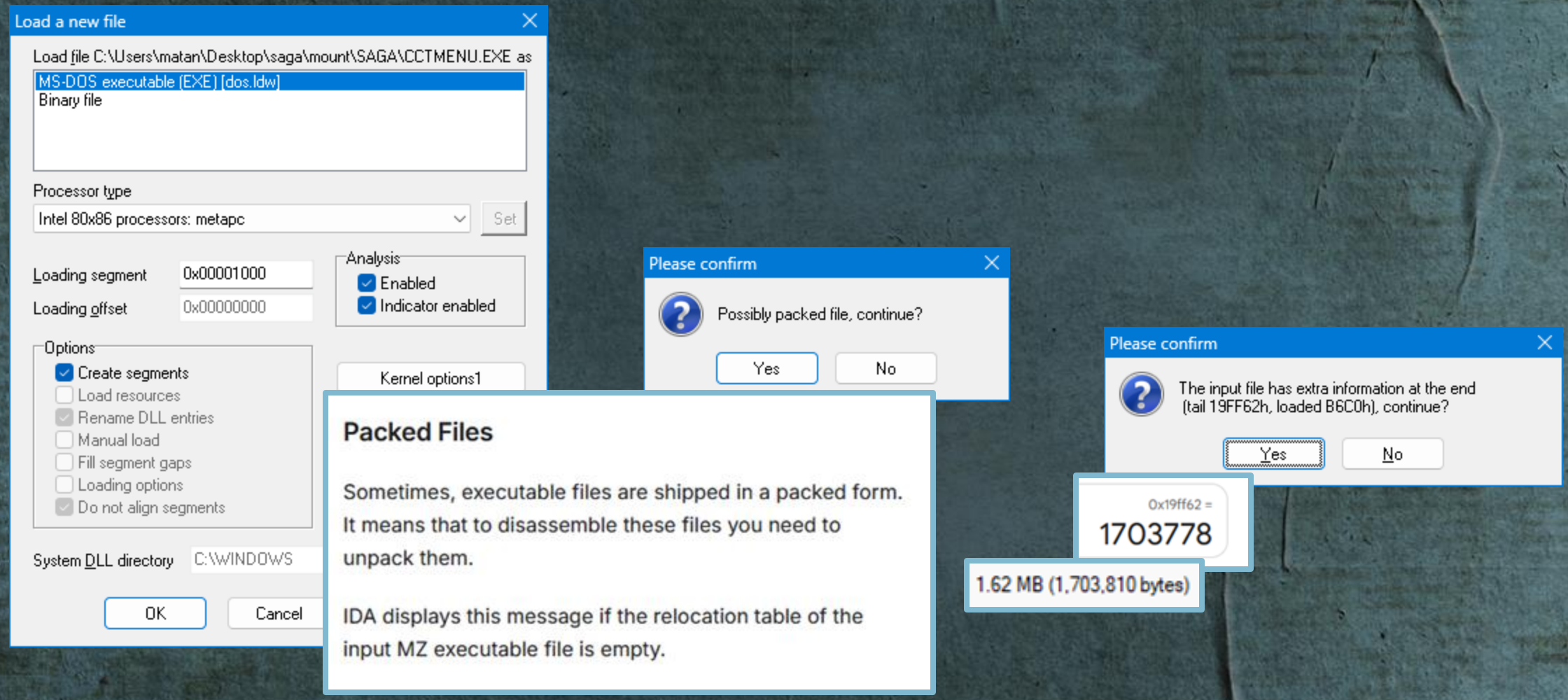

COMPLEX

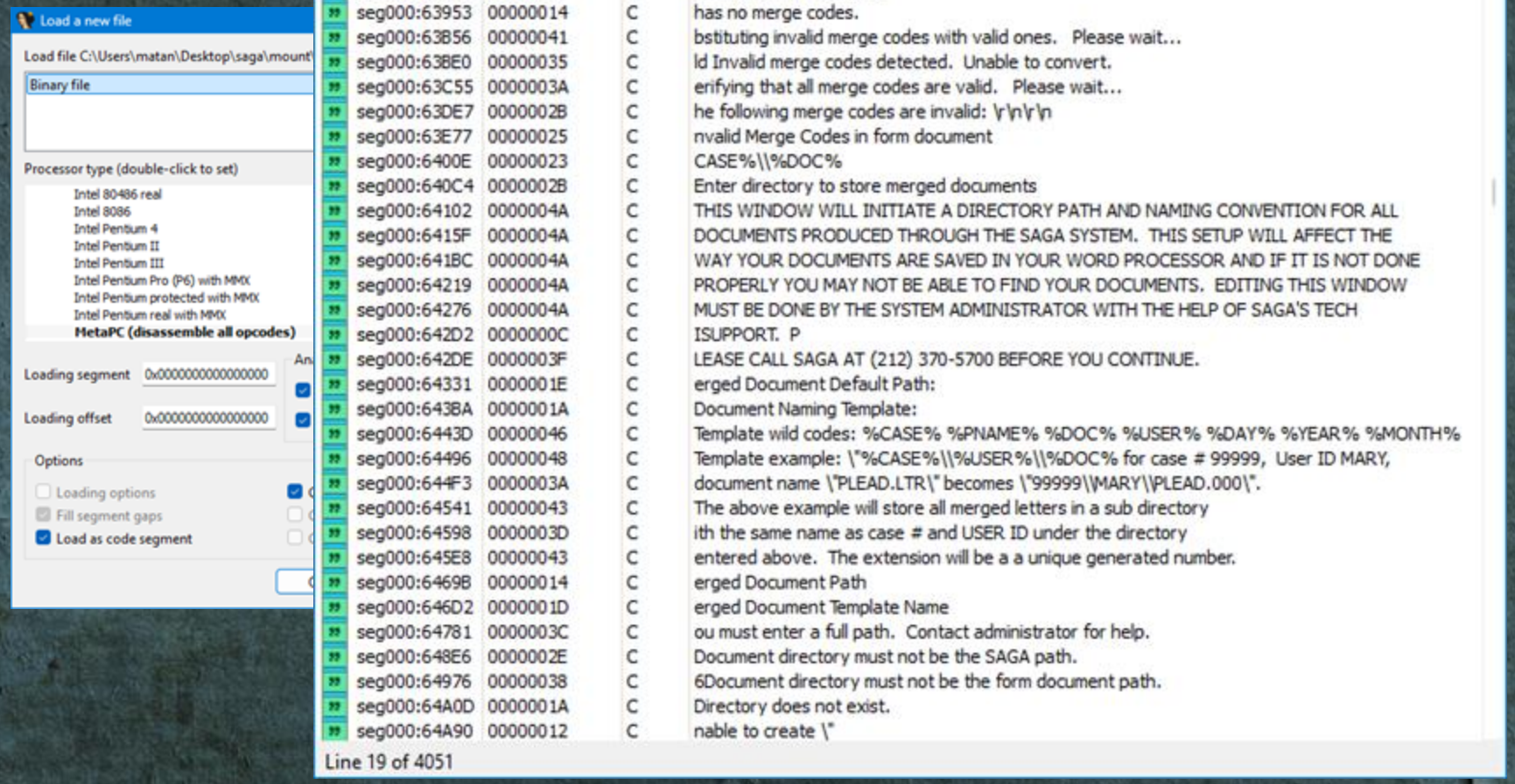

WHEN?
WHAT?
WHERE?
WHY?
WHO?
HOW?


HARD
EASY


FIND A
SOLUTION


ARCHIVE
FORMBOOK
PIDEMO
SFORMS
TEMP
CASEDEL.EXE
CASEREST.EXE
CASEUND.EXE
1.62 MB
CCTMENU.EXE
DDMGR.EXE
DSALVAGE.EXE
FA.EXE
FLOATERS.EXE
FTEST.EXE
LS.EXE
LTRHREST.EXE
NOTEAN.EXE
PKUNZIP.EXE
PKZIP.EXE
Q.EXE
RESTMEM.EXE
SETPATH.EXE
WINSTART.EXE
WPFIELDS.EXE
XD.EXE
XD.CFG
ARCHIVE.DBF
CAL.DBF
CASE.DBF
CASEEXP.DBF
CATYFIRM.DBF
CEXPWIT.DBF
CHANGEID.DBF
DIRLITIG.DBF
DIRMERGE.DBF
DIRPRINT.DBF
DIRPROV.DBF
DIRSL.DBF
DIRSTAFF.DBF
DIRSTAT.DBF
DIRTASKS.DBF
DOCTORS.DBF
DWRPART.DBF
EXPGRP.DBF
EXPTYPE.DBF
EXPWGRP.DBF
EXPWIT.DBF
NEGOHIST.DB
NOFAULT.DBF
OTHERS.DBF
OTHRPT.DBF
PIPORT.DBF
PKFIELDS.DBF
PKHEDULE.DB
PKINDEX.DBF
PKRMERGE.DBF
POLICE.DBF
PROPDEM.DBF
REPAIR.DBF
RPTTEMP.DBF
RRDD.DBF
CHG.ICO
MOVE2.ICO
PIDEMO.ICO
SAGA.ICO
SAGA.INI
SAGABAK0.INI
SAGA.LCK
TTSAGA.LOG
READ.ME
SAGA.GRP
About
IDA - The Interactive Disassembler
Freeware Version 5.0
(c) 2010 Hex-Rays SA
Welcome to the freeware edition of IDA Pro 5.0.
This version is fully functional but does not offer all the bells and whistles of the commercial versions of IDA Pro.
Try the commercial version of IDA Pro today!
http://www.hex-rays.com
Do not display IDA 6.x info next time
OK
SE III PLUS
DE
- A database system
- A compiler
ASHTON-TATE


Load a new file
Load file C:\Users\matan\Desktop\saga\mount\SAGA\CCTMENU.EXE as
MS-DOS executable (EXE) [dos.ldw]
Binary file
Processor type
Intel 80x86 processors: metapc
Set
Loading segment
0x00001000
Loading offset
0x00000000
Analysis
Enabled
Indicator enabled
Options
Create segments
Load resources
Rename DLL entries
Manual load
Fill segment gaps
Loading options
Do not align segments
Kernel options1
System DLL directory
C:\WINDOWS
OK
Cancel
Please confirm
Possibly packed file, continue?
Yes
No
Please confirm
The input file has extra information at the end
(tail 19FF62h, loaded B6C0h), continue?
Yes
No
0x19ff62 =
1703778
1.62 MB (1,703,810 bytes)
Packed Files
Sometimes, executable files are shipped in a packed form. It means that to disassemble these files you need to unpack them.
IDA displays this message if the relocation table of the input MZ executable file is empty.

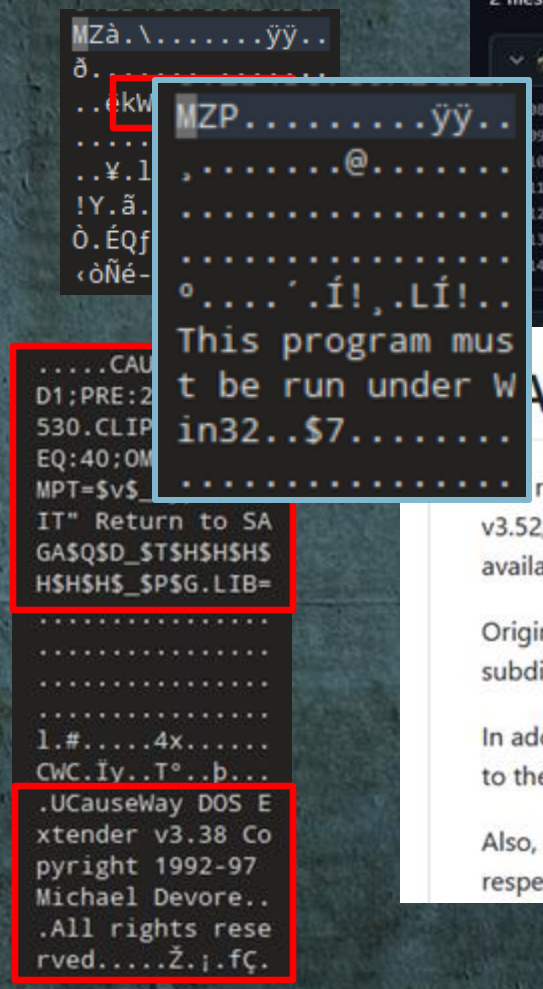


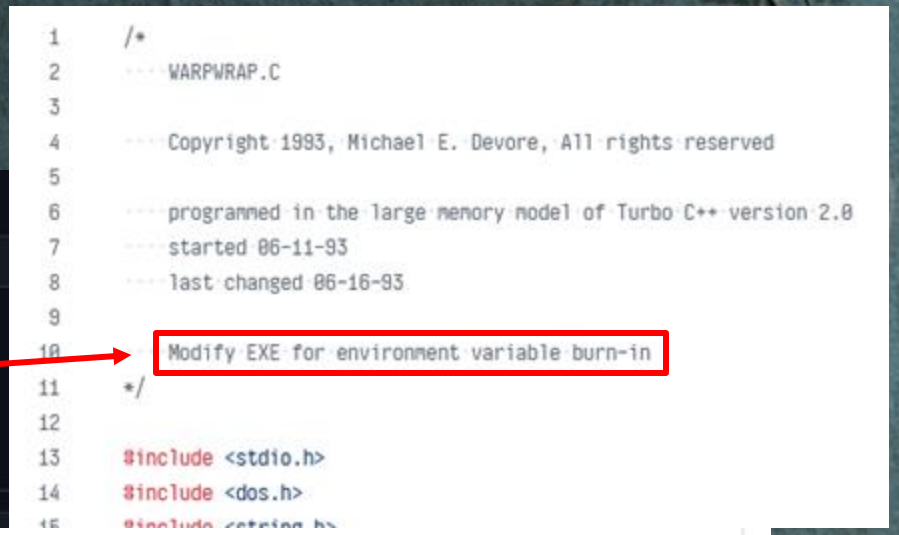
Strings

Address	Length	Type	String
seg000:6387A	00000035	C	etrieving form document merge codes. Please wait...
seg000:638D2	00000035	C	nalyzing form document merge codes. Please wait...
seg000:63935	00000017	C	arning: Form document
seg000:63953	00000014	C	has no merge codes.
seg000:63B56	00000041	C	bstituting invalid merge codes with valid ones. Please wait...
seg000:63BE0	00000035	C	ld Invalid merge codes detected. Unable to convert.
seg000:63C55	0000003A	C	erifying that all merge codes are valid. Please wait...
seg000:63DE7	0000002B	C	he following merge codes are invalid: \r\n\r\n
seg000:63E77	00000025	C	nvalid Merge Codes in form document
seg000:6400E	00000023	C	CASE%\\%DOC%
seg000:640C4	0000002B	C	Enter directory to store merged documents
seg000:64102	0000004A	C	THIS WINDOW WILL INITIATE A DIRECTORY PATH AND NAMING CONVENTION FOR ALL
seg000:6415F	0000004A	C	DOCUMENTS PRODUCED THROUGH THE SAGA SYSTEM. THIS SETUP WILL AFFECT THE
seg000:641BC	0000004A	C	WAY YOUR DOCUMENTS ARE SAVED IN YOUR WORD PROCESSOR AND IF IT IS NOT DONE
seg000:64219	0000004A	C	PROPERLY YOU MAY NOT BE ABLE TO FIND YOUR DOCUMENTS. EDITING THIS WINDOW
seg000:64276	0000004A	C	MUST BE DONE BY THE SYSTEM ADMINISTRATOR WITH THE HELP OF SAGA'S TECH
seg000:642D2	0000000C	C	ISUPPORT. P
seg000:642DE	0000003F	C	LEASE CALL SAGA AT (212) 370-5700 BEFORE YOU CONTINUE.
seg000:64331	0000001E	C	erged Document Default Path:
seg000:643BA	0000001A	C	Document Naming Template:
seg000:6443D	00000046	C	Template wild codes: %CASE% %PNAME% %DOC% %USER% %DAY% %YEAR% %MONTH%
seg000:64496	00000048	C	Template example: \"%CASE%\\%USER%\\%DOC% for case # 99999, User ID MARY,
seg000:644F3	0000003A	C	document name \"PLEAD.LTR\" becomes \"99999\\MARY\\PLEAD.000\".
seg000:64541	00000043	C	The above example will store all merged letters in a sub directory
seg000:64598	0000003D	C	ith the same name as case # and USER ID under the directory
seg000:645E8	00000043	C	entered above. The extension will be a a unique generated number.
seg000:6469B	00000014	C	erged Document Path
seg000:646D2	0000001D	C	erged Document Template Name
seg000:64781	0000003C	C	ou must enter a full path. Contact administrator for help.
seg000:648E6	0000002E	C	Document directory must not be the SAGA path.
seg000:64976	00000038	C	6Document directory must not be the form document path.
seg000:64A0D	0000001A	C	Directory does not exist.
seg000:64A90	00000012	C	nable to create \"

Line 19 of 4051

Binary Analysis





AUSEWAY DOS EXTENDER

repository contains the source code and binaries of Devore Software & Consulting's CauseWay DOS Extender v3.52, released 2000-01, for the purpose of archival and preservation. As of 2022-01, the original web site is still available, but the pages containing the software are not; a mirror is available on the Wayback Machine.

Originally, the code was in a subdirectory under 'source/all'. Everything has been moved up one directory and the 'all' subdirectory has been removed. The 'source' directory has been renamed 'src'.

In addition to the binaries and source code, a copy of the CauseWay documentation in HTML format has been added to the 'doc' subdirectory.

Also, the source code for WarpLink v2.70 and WarpLink32 have been included under 'src/wl' and 'src/wl32', respectively.

DOS Extenders

- DOS stub -> switch into protected mode -> run protected-mode program

- Protected mode enables
  - More usable memory (640 KB → 4 GB on 386+ CPUs)
  - Natural use of newer, 32-bit CPUs (>=386)
  - Flat 32-bit memory addressing

- Examples: DOS/4GW, CWSDPMI, Phar Lap…
- Used in games, CAD, database applications

Jazz Jackrabbit (1994) is built with DOS/4GW

CauseWay DOS Extender

- **Stub is linked into the final EXE, runs embedded executable**

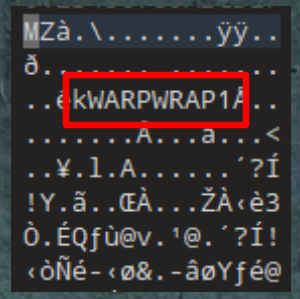
- Supports running in both 16- and 32-bit protected mode
- Supports very large memory models (>80386SX)
- Compatible with VCPI / DPMI
- Makes full use of 386-level chip capabilities

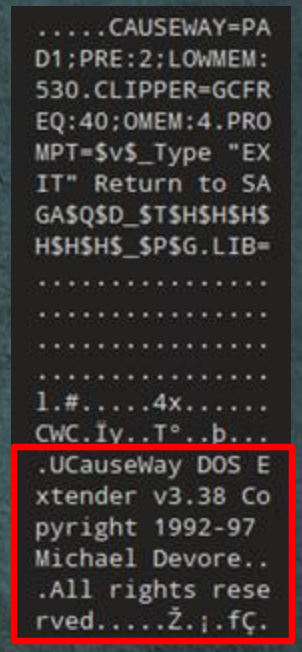
- Tooling included –
  - Linker (WarpLink)
  - Debugger (CWD)
  - Support & enhancement utilities

```
;-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
;
;Main code segment. This takes care of calling the right initialisation routines
;and generaly getting everything rolling.
;
_cwMain segment para public 'Main thread' use16
    assume cs:_cwMain, ds:_cwMain, ss:_cwStack
;
;Want a copyright message embedded first.
;
Copyright   label byte
    db 'CauseWay DOS Extender v'
VersionMajor    db '3.'
VersionMinor    db '49'
    db " Copyright 1992-99 Michael Devore.",13,10,"All rights reserved.",13,10,0
```

```
l.#.....4x......
CWC.Ïy..T°..b...
.UCauseWay DOS E
xtender v3.38 Co
pyright 1992-97
Michael Devore..
.All rights rese
rved.....Ž.¡.fÇ.
```

Binary Analysis



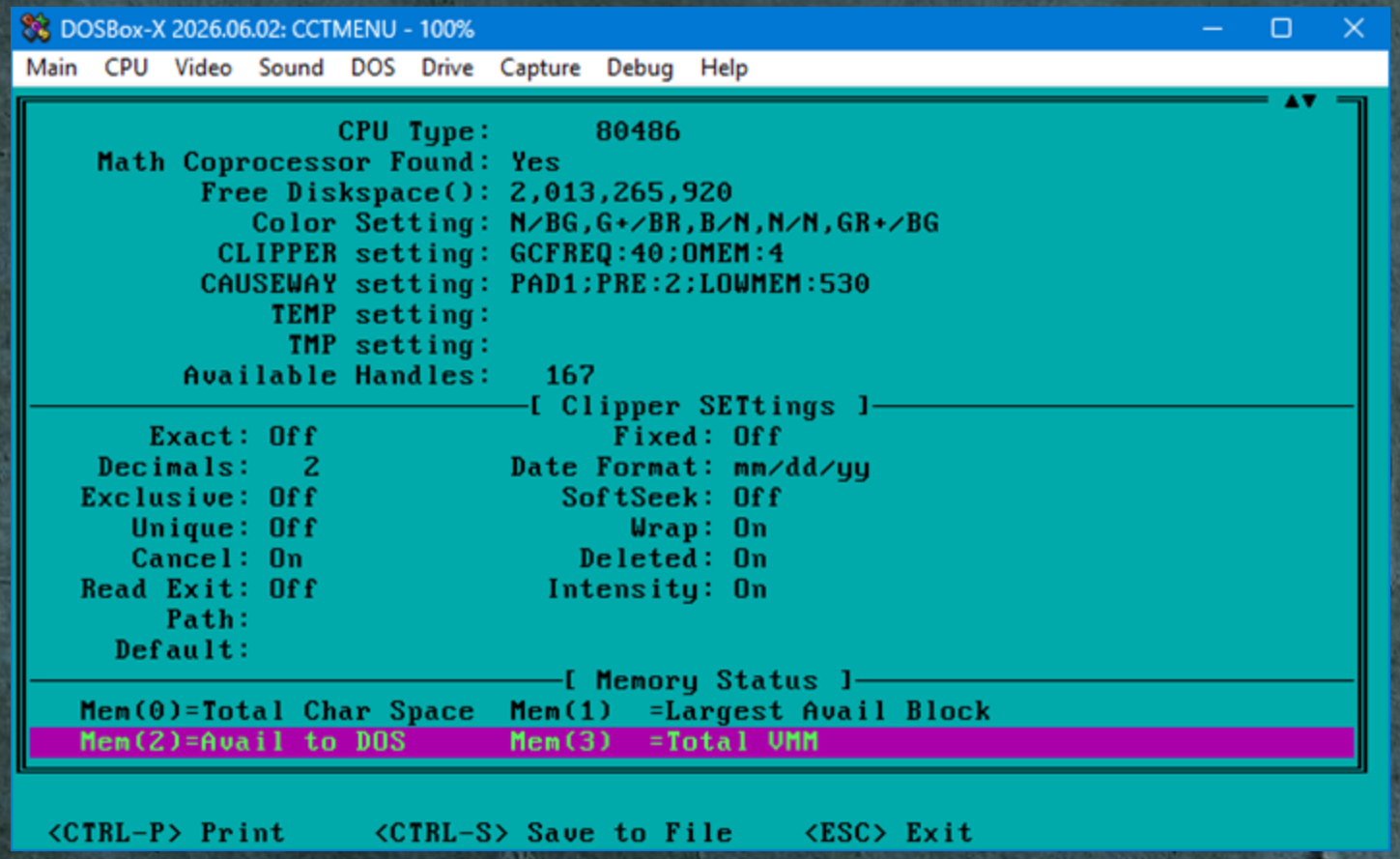


CauseWay 3P executable (32-bit protected mode) ↴

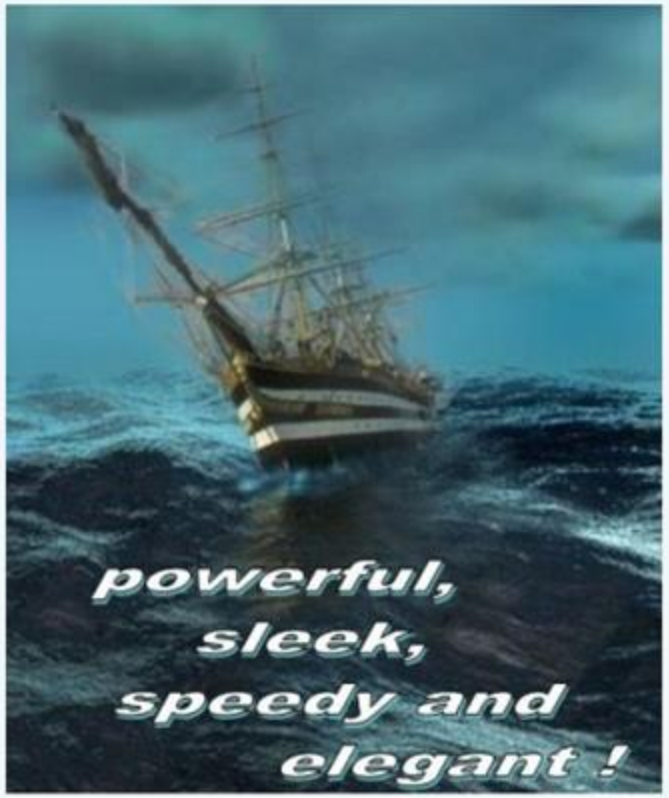
33	50	61	44	19	00	71	18	18	00	08	35	19	00	CA	0A	3PaD..q....5..Ê.
58	35	00	00	10	00	00	00	FC	05	58	1B	00	00	C9	0A	X5......ü.X...É.
01	40	00	40	00	00	00	00	00	00	00	00	00	00	00	00	.@.@............
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00	00	00	00	5A	12	00	02	60	12	00	00	47	00	00	02	....Z...`...G...
B0	12	00	00	F8	00	00	02	B0	13	00	00	F6	00	00	02	°...ø...°...ö...
B0	14	00	00	93	02	00	02	50	17	00	00	16	02	00	02	°...“...P.......
70	19	00	00	C0	01	00	02	30	1B	00	00	CA	06	00	02	p...À...0...Ê...
00	22	00	00	15	06	00	02	20	28	00	00	AB	01	00	02	."...... (..«...
D0	29	00	00	9E	04	00	02	70	2E	00	00	52	01	00	02	Ð)..ž...p...R...
D0	2F	00	00	36	00	00	02	10	30	00	00	10	07	00	02	Ð/..6....0......
20	37	00	00	B0	03	00	02	D0	3A	00	00	9D	06	00	02	7..°...Ð:.......
70	41	00	00	CA	02	00	02	40	44	00	00	B0	04	00	02	pA..Ê...@D..°...
F0	48	00	00	59	06	00	02	50	4F	00	00	DD	00	00	02	ðH..Y...PO..Ý...
30	50	00	00	1E	00	00	02	50	50	00	00	48	03	00	02	0P......PP..H...
A0	53	00	00	31	02	00	02	E0	55	00	00	CD	2F	00	02	S..1...àU..Í/..
B0	85	00	00	D1	05	00	02	90	8B	00	00	33	1D	00	02	°….Ñ....‹..3...

Crossroads – How to Continue?

A Different Approach



Why Clipper ?



Why not ? Clipper is a high level programming language and is the best one.

```
COMMAND - DOS in a BOX
C:\test>copy world.prg con
world.prg => con
?"Hello world!"
C:\test>clipper world
Clipper (R) Version 5.01
Copyright (c) Nantucket Corp 1985-1991.  All Rights Reserved.
Microsoft C Floating Point Support Routines
Copyright (c) Microsoft Corp 1984-1987.  All Rights Reserved.
374K available
Compiling WORLD.PRG
Code size 51, Symbols 48, Constants 13
C:\test>rtlink file world
.RTLink for Clipper  Dynamic Overlay Linker / Prc Linker  Version 3.13
(C) Copyright Pocket Soft Inc., 1988-1991.  All Rights Reserved.

134K
C:\test>dir/b
WORLD.EXE
WORLD.PRG
WORLD.OBJ
C:\test>world

Hello world!
C:\test>
```

Compiling and running hello world program
From Wikipedia, the free encyclopedia

- xBase compiler
- Created in 1985 as a replacement for dBASE III
- Usage of Clipper declined in the early 90's
- Newer implementations actively developed to this day!

Date & Time

SET CENTURY	Modify the date format to include or omit century digits
SET DATE	Set the date format for input and display
SET EPOCH	Control the interpretation of dates with no century digits

Set()

Posted on January 19, 2014 by vivaclipper

Set()

Changes or evaluated environmental settings

Syntax

```
Set(<nSet> [,  <xNewSetting> [,  <xOption> ] ] ) --> xPreviousSetting
```

Arguments

<nSet> Set Number

<xNewSetting> Any expression to assign a value to the setting

<xOption> Logical expression

<nSet> <xNewSetting> <xOption>

_SET_EPOCH <nYear>

Determines how to handle the conversion of 2-digit years to 4 digit years. When a 2-digit year is greater than or equal to the year part of the epoch, the century part of the epoch is added to the year. When a 2-digit year is less than the year part of the epoch, the century part of the epoch is incremented and added to the year. The default epoch is 1900, which converts all 2-digit years to 19xx. Example: If the epoch is set to 1950, 2-digit years in the range from 50 to 99 get converted to 19xx and 2-digit years in the range 00 to 49 get converted to 20xx.

Let's search 0x786 (1926) in binary, see what we find:

```
01:E2D0  B8 16 00 BB F9 09 53 BB 00 00 53 0E 50 9A 32 01  ¸...»ù.S»..S.Pš2.
01:E2E0  F8 05 83 C4 08 CB 2A 00 00 28 08 09 10 08 00 2A  ø.ƒÄ.Ë*..(.....*
01:E2F0  15 00 72 72 64 2F 0E 00 2F 0A 00 2F 09 00 2A 17  ..rrd/../../..*.
01:E300  00 64 64 01 05 58 58 58 58 58 00 2F 11 00 2F 10  .dd..XXXXX./../.
01:E310  00 2F 0F 00 2A 1C 00 3B 14 00 2B 01 00 71 71 12  ./..*..;..+..qq.
01:E320  02 00 12 03 00 12 04 00 12 05 00 12 06 00 12 07  ................
01:E330  00 12 09 00 33 07 00 07 05 00 07 04 00 07 02 00  ....3...........
01:E340  2A 20 00 13 0A 00 3B 05 00 3B 86 07 27 02 00 2A  * ....;..;†.'..*
01:E350  22 00 13 0B 00 13 0C 00 29 00 00 27 01 00 2A 38  "........)..'..*8
01:E360  00 13 0D 00 29 00 00 70 1B 70 00 2A 39 00 13 0E  ....)..p.p.*9...
01:E370  00 3B 90 01 3B 05 00 27 02 00 2A 3A 00 13 0F 00  .;..;..'..*:....
01:E380  29 00 00 7C C8 76 13 10 00 01 3C 52 65 69 6E 64  )..|Èv....<Reind
01:E390  65 78 2F 4D 61 69 6E 74 65 6E 61 6E 63 65 20 69  ex/Maintenance i
01:E3A0  6E 20 70 72 6F 67 72 65 73 73 2E 2E 2E 20 50 6C  n progress... Pl
```

Find Results

Address	Value
Found 12 occurrences of '86 07'.	
8E55h	86 07
14AB4h	86 07
1E34Ah	86 07
646B6h	86 07
77741h	86 07
B2605h	86 07
B60EBh	86 07
BD705h	86 07
CC2FBh	86 07
DAFD1h	86 07
114EE6h	86 07
18AA5Eh	86 07

Inspector

Type	Value
Binary	10000110
Signed Byte	-122
Unsigned Byte	134
Signed Short	1926
Unsigned Short	1926
Signed Int	36112262
Unsigned Int	36112262
Signed Int64	9616328732641158
Unsigned Int64	9616328732641158
Float	1.227138e-37
Double	5.05204761481378e-308
Half Float	0.0001147985
String	†□'□
DOSDATE	12/06/1983

Original

Patched

DOSBox-X 2026.06.02: CCTMENU - 100%

Main CPU Video Sound DOS Drive Capture Debug Help

06/08/26 NIMA AT LAW (DOG LAWYER) GMBH CCT7100
Calendar Entry ADMIN

[Limit View for Calendar Listing]

Attorney :
Case Atty:
Judge Id :
Category :
County :
Location :
Caption :

From Date: 01/01/26 Friday

Status: Outstanding

DOSBox-X 2026.06.02: CCTMENU - 100%

Main CPU Video Sound DOS Drive Capture Debug Help

06/08/26 NIMA AT LAW (DOG LAWYER) GMBH CCT7100
Calendar Entry ADMIN

[Limit View for Calendar Listing]

Attorney :
Case Atty:
Judge Id :
Category :
County :
Location :
Caption :

From Date: 01/01/26 Thursday

Status: Outstanding

January 2026

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3

THANK YOU

BOTANICA
SOFTWARE LABS