The deep dive into the world of

viruses

Ben Cartwright-Cox
>>>@benjojo12<<<

Age of MS-DOS:

Age of Presenter:

Age of MS-DOS: (1988 for 4.0)

Age of Presenter: (1995 barely stable)


Anyway

What is DOS?

Marcin Wichary - CC BY 2.0

Nick Bair - CC BY-SA 2.0

- DOS is a upgrade from CP/M

- DOS is a upgrade from CP/M
- DOS covers a wide range of vendors

- DOS is a upgrade from CP/M
- DOS covers a wide range of vendors
- Some of them had some compatibility with each other

- DOS is a upgrade from CP/M
- DOS covers a wide range of vendors
- Some of them had some compatibility with each other
- Meaning some DOS's shared malware with each other

TECHNO

VX Heavens

Hosted sites
eof-project.net
bi0tic.info
29a
Berniee
Bull Moose
DCA
Doomriderz
IKX
fAMINE
herm1t
Positron
V-Codez
WarGame
Friendly sites
FreeThinking
Scut Anti Virus
Offensive Computing
You link here?

Viruses don't harm, ignorance does!

"Everyone has the right to freedom of opinion and expression; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas througt any media and regardless of frontiers."
Article 19 of "Universal Declaration of Human Rights"

Welcome to VX Heavens! This site is dedicated to providing information about *computer viruses* (or *virii*, as some would prefer) to anyone who is interested in this topic.

This site contains a massive, continuously updated collection of magazines, virus samples, virus sources, polymorphic engines, virus generators, virus writing tutorials, articles, books, news archives etc. Even the viruses for the platforms you've never heard of. We also offer free hosting for virus authors and groups.

Some of you might reasonably say that it is illegal to offer such content on the net. Or that this information can be misused by "malicious people". I only want to ask that person: *"Is ignorance a defence?"*

webmaster@vx.netlux.org

You can help us improve the site by uploading new stuff, making a donation, posting our link to your site, blog or forum, or leaving a comment. Thank you!

What's new? (April 2010)

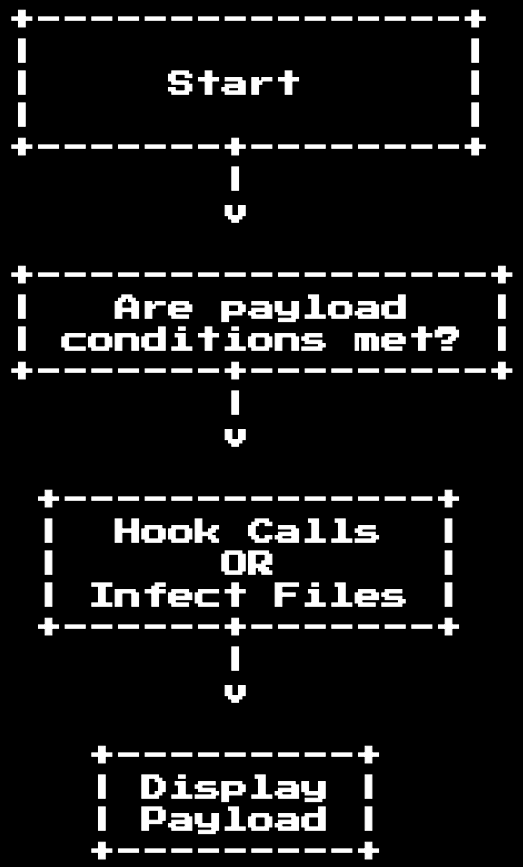
29 + DL/PE: DTrash 1.0
29 + DL/PE: 451's Dirty Entry Point Engine (DEE 1.32)
28 + DL/MAG: The Scanner
28 + DL/GEN: WAH C# Worm Generator (WAH 1.0) by Retro
23 + LIB/EN: Z0mbie "The virmaking is dying"

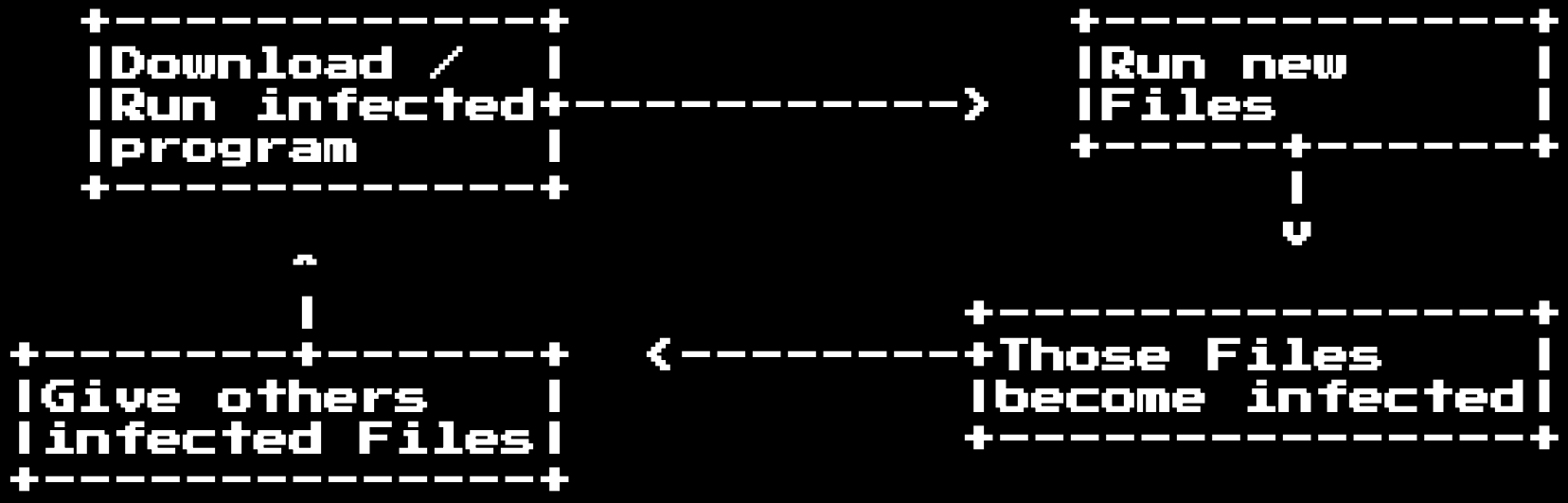
Friday, 23 March, the server has being seized by the police forces due to the criminal investigation (article 361-1 Criminal Code of Ukraine - the creation of the malicious programs with an intent to sell or spread them) based on someone's tip-off on "placement into the free access malicious software designed for the unauthorized breaking into computers, automated systems, computer networks".

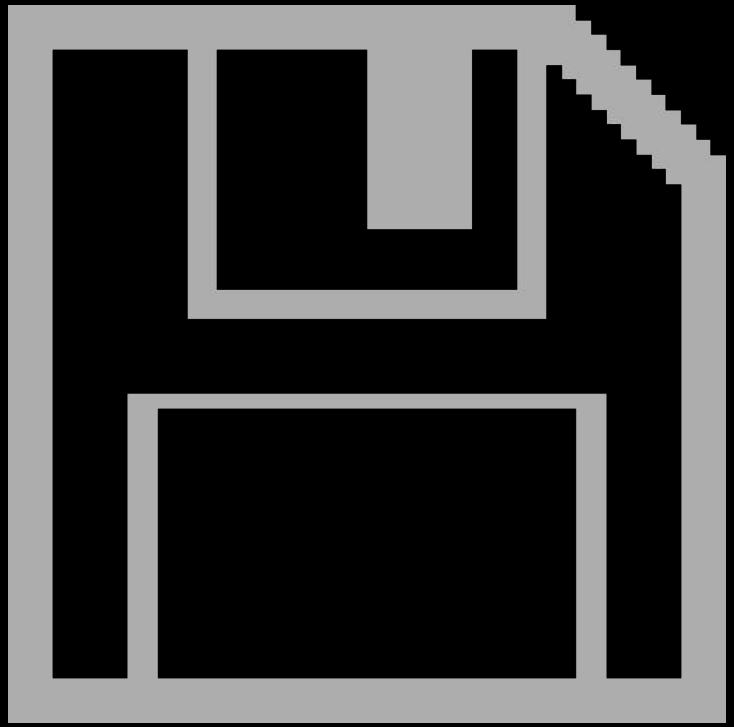
```
$ tar -tvf viruses-20070914.tar | wc -l
66714


$ ls -alh viruses-20070914.tar
6.6G viruses-20070914.tar
```

AMBULANCE


Start
Are payload conditions met?
Hook Calls
OR
Infect Files
Display Payload


Download / Run infected program
Run new Files
Those Files become infected
Give others infected Files

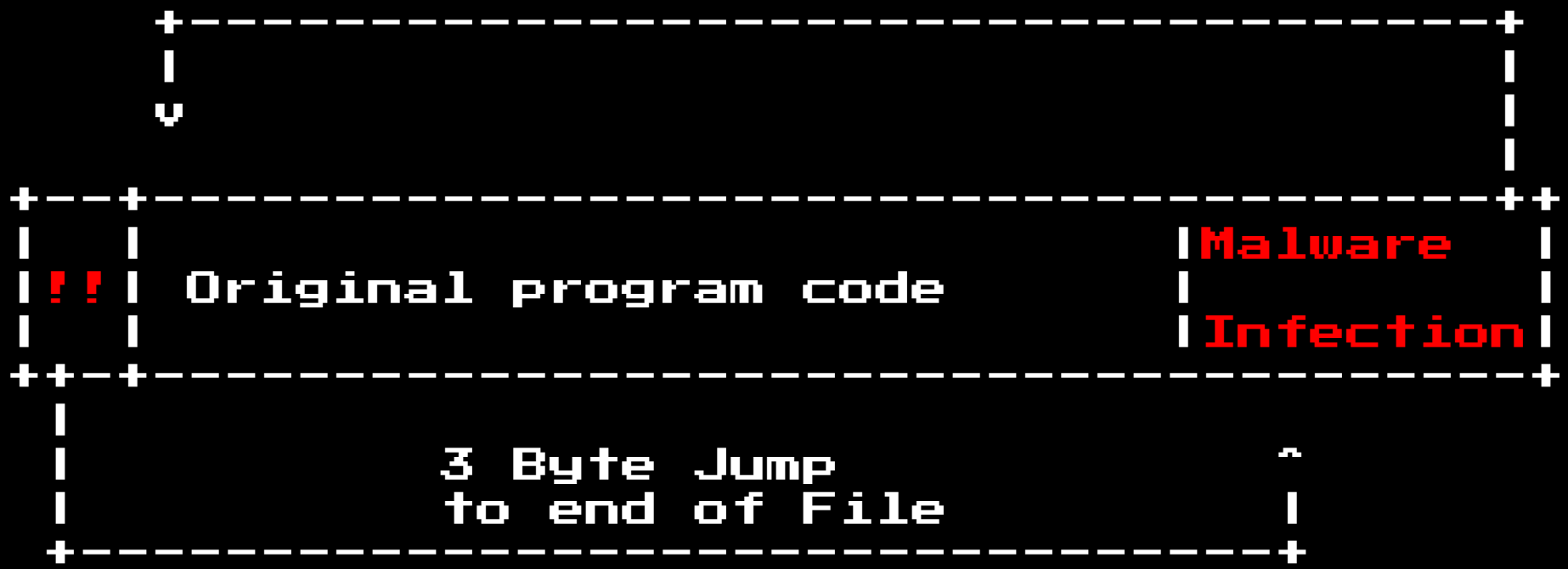



Start
Are payload conditions met?
Hook Calls OR Infect Files
Display Payload

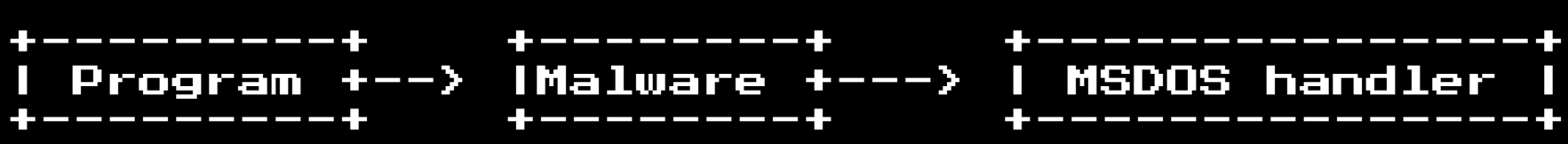

Start
Are payload conditions met?
Hook Calls OR Infect Files
Display Payload


Start
Are payload conditions met?
Hook syscalls OR Scan for Files To Infect
Display Payload


Original program code


!!
Original program code
Malware
Infection
3 Byte Jump
to end of File


Program
MSDOS handler


Program
Malware
MSDOS handler

Int 21h Syscalls

0 - Terminate

1 - Keyboard input

2 - Display output

3 - Wait for device input

...

9 - Print string

F - Open File

2A - Get Date

2C - Get Time

31 - TSR

40 - Write to File

41 - Delete File

48 - Allocate RAM

4C - Exit with return code

```
[org 100h]
mov dx,msg
mov ah,9
int 21h

mov ah,4Ch
int 21h
msg db 'Hello, World!',0Dh,0Ah,'$'
```

Int 21h Syscalls

0 - Terminate

1 - Keyboard input

2 - Display output

3 - Wait for device input

...

9 - Print string

F - Open File

2A - Get Date

2C - Get Time

31 - TSR

40 - Write to File

41 - Delete File

48 - Allocate RAM

4C - Exit with return code

```
[org 100h]
mov dx,msg
mov ah,9
int 21h

mov ah,4Ch
int 21h
msg db 'Hello, World!',0Dh,0Ah,'$'
```

Int 21h Syscalls

0 - Terminate

1 - Keyboard input

2 - Display output

3 - Wait for device input

...

9 - Print string

F - Open File

2A - Get Date

2C - Get Time

31 - TSR

40 - Write to File

41 - Delete File

48 - Allocate RAM

4C - Exit with return code

Int 21h Syscalls

0 - Terminate

1 - Keyboard input

2 - Display output

3 - Wait for device input

...

9 - Print string

F - Open File

2A - Get Date

2C - Get Time

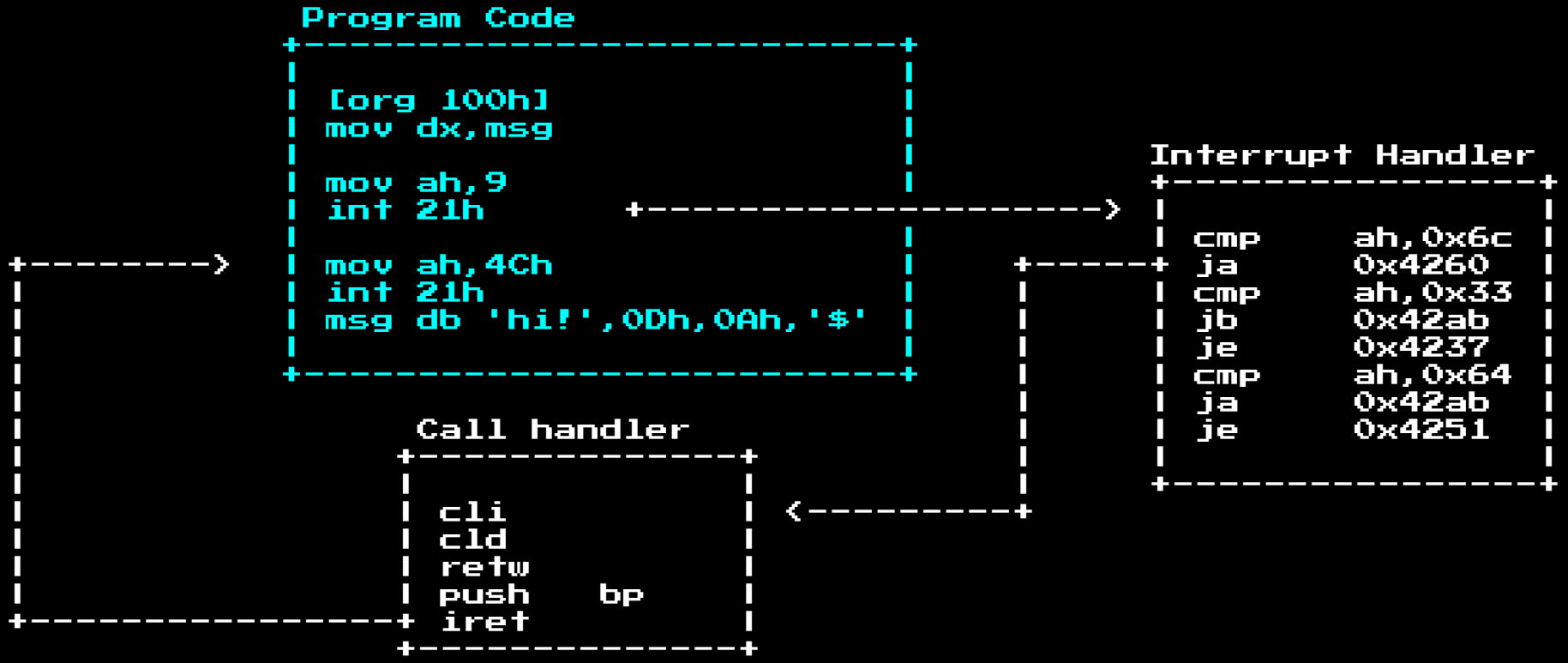
31 - TSR

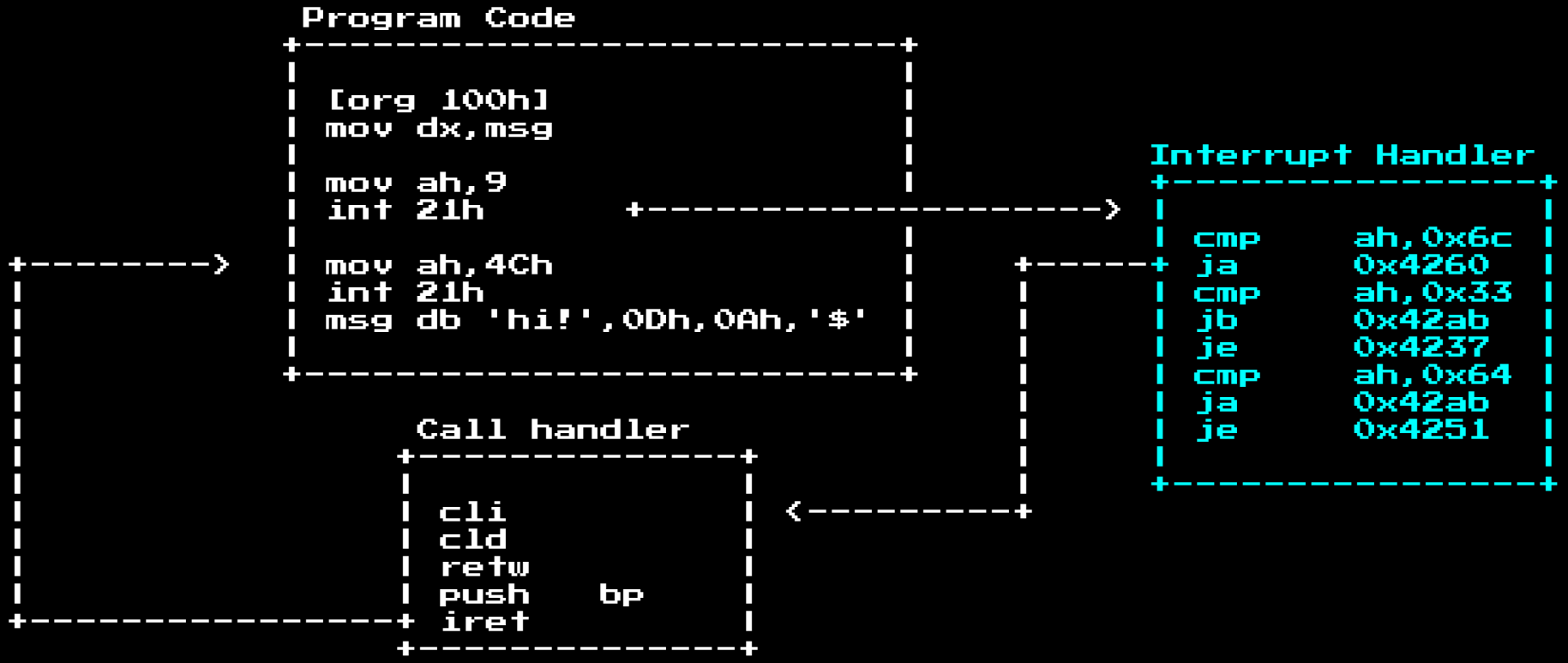
40 - Write to File

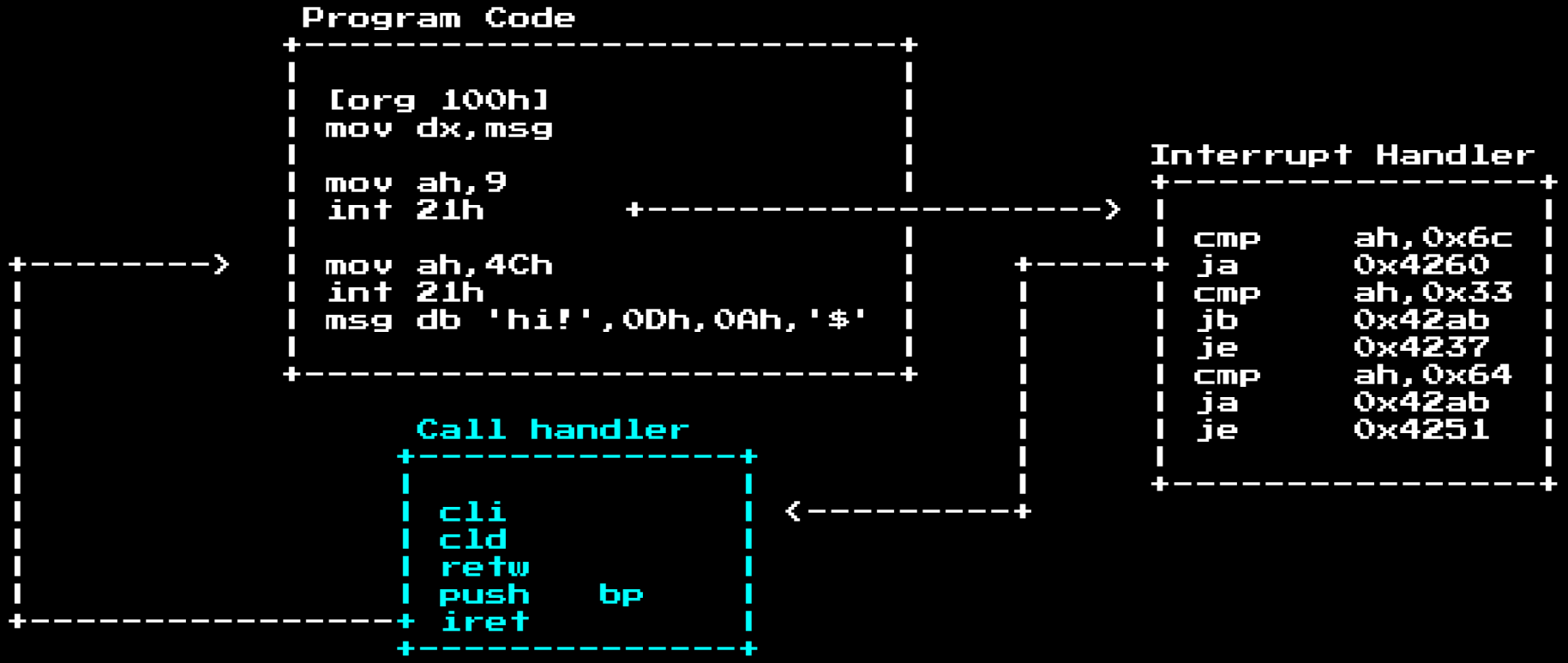
41 - Delete File

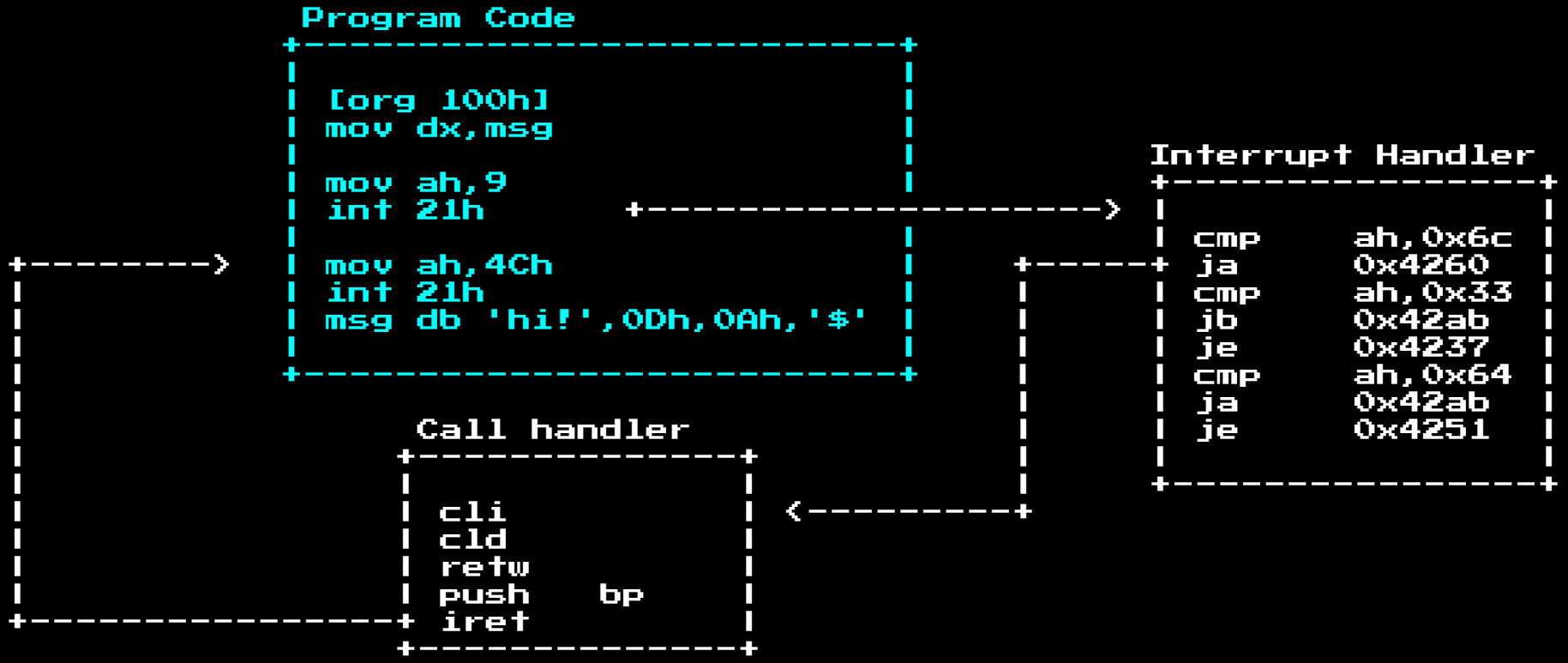
48 - Allocate RAM

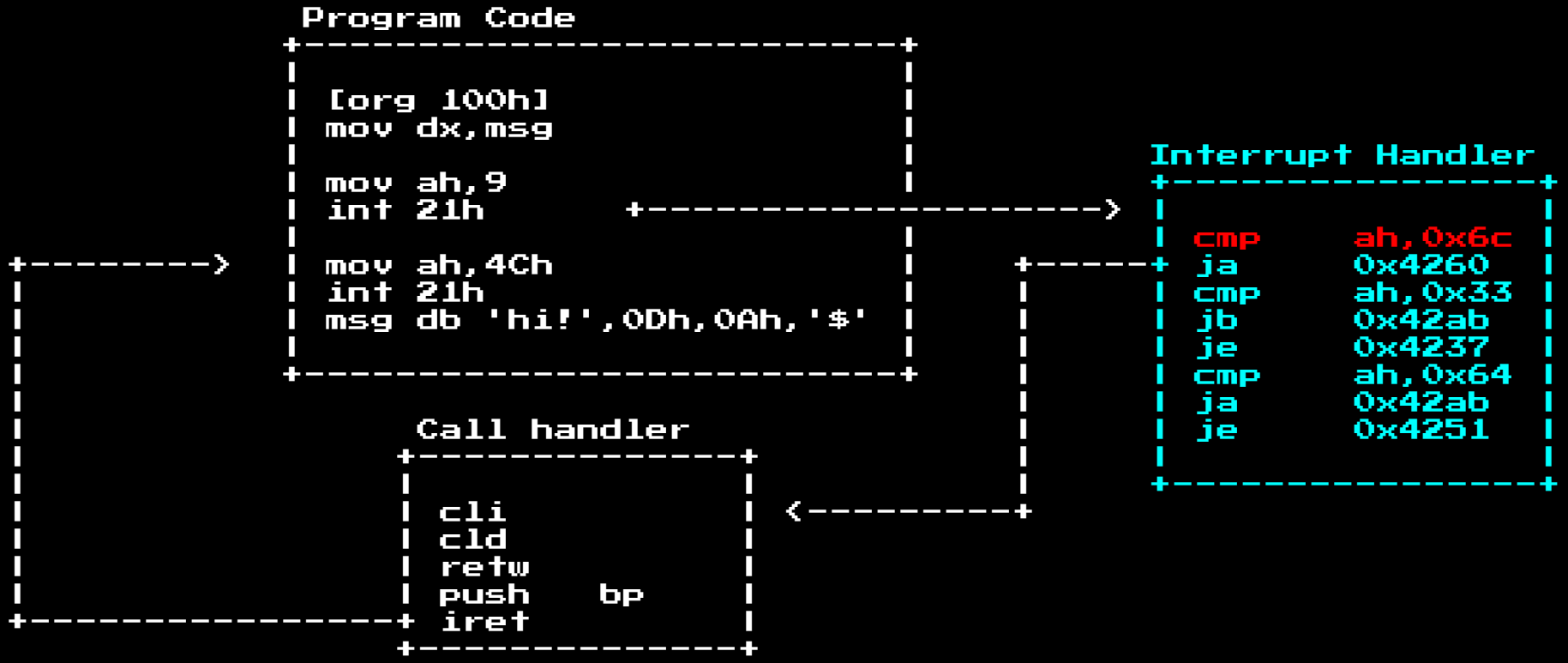
4C - Exit with return code


Program Code
[org 100h]
mov dx,msg
mov ah,9
int 21h
mov ah,4Ch
int 21h
msg db 'hi!',0Dh,0Ah,'$'
Interrupt Handler
cmp ah,0x6c
ja 0x4260
cmp ah,0x33
jb 0x42ab
je 0x4237
cmp ah,0x64
ja 0x42ab
je 0x4251
Call handler
cli
cld
retw
push bp
iret


Program Code
[org 100h]
mov dx,msg
mov ah,9
int 21h
mov ah,4Ch
int 21h
msg db 'hi!',0Dh,0Ah,'$'
Interrupt Handler
cmp ah,0x6c
ja 0x4260
cmp ah,0x33
jb 0x42ab
je 0x4237
cmp ah,0x64
ja 0x42ab
je 0x4251
Call handler
cli
cld
retw
push bp
iret


Program Code
[org 100h]
mov dx,msg
mov ah,9
int 21h
mov ah,4Ch
int 21h
msg db 'hi!',0Dh,0Ah,'$'
Interrupt Handler
cmp ah,0x6c
ja 0x4260
cmp ah,0x33
jb 0x42ab
je 0x4237
cmp ah,0x64
ja 0x42ab
je 0x4251
Call handler
cli
cld
retw
push bp
iret


Program Code
[org 100h]
mov dx,msg
mov ah,9
int 21h
mov ah,4Ch
int 21h
msg db 'hi!',0Dh,0Ah,'$'
Interrupt Handler
cmp ah,0x6c
ja 0x4260
cmp ah,0x33
jb 0x42ab
je 0x4237
cmp ah,0x64
ja 0x42ab
je 0x4251
Call handler
cli
cld
retw
push bp
iret


Program Code
[org 100h]
mov dx,msg
mov ah,9
int 21h
mov ah,4Ch
int 21h
msg db 'hi!',0Dh,0Ah,'$'
Interrupt Handler
cmp ah,0x6c
ja 0x4260
cmp ah,0x33
jb 0x42ab
je 0x4237
cmp ah,0x64
ja 0x42ab
je 0x4251
Call handler
cli
cld
retw
push bp
iret

Syscall Op	Syscall Name
48	Get DOS version
61	Open file
66	Move file pointer
63	Read file or device (Read 2 bytes on handle 5)
62	Close file
48	Get DOS version
61	Open file
66	Move file pointer
63	Read file or device (Read 6800 bytes on handle 5)
66	Move file pointer
64	Write file or device (Write 6800 bytes on handle 5)
66	Move file pointer
64	Write file or device (Write 0 bytes on handle 5)
87	Get or set file date and time
62	Close file
48	Get DOS version
41	Parse filename
41	Parse filename
75	Execute program
9	Display string
76	Terminate with return code (Return code = '36')
64	Write file or device (Write 0 bytes on handle 1)

```
Syscall Op   |  Syscall Name
48    Get DOS version
61    Open file
66    Move file pointer
63    Read file or device (Read 2 bytes on
handle 5)
62    Close file
48    Get DOS version
61    Open file
66    Move file pointer
63    Read file or device (Read 6800 bytes on
handle 5)
66    Move file pointer
64    Write file or device (Write 6800 bytes on
handle 5)
66    Move file pointer
64    Write file or device (Write 0 bytes on
handle 5)
87    Get or set file date and time
62    Close file
48    Get DOS version
41    Parse filename
41    Parse filename
75    Execute program
9     Display string
76    Terminate with return code (Return code
= '36')
64    Write file or device (Write 0 bytes on
handle 1)
```

Syscall Op	Syscall Name
48	Get DOS version
61	Open file
66	Move file pointer
63	Read file or device (Read 2 bytes on handle 5)
62	Close file
48	Get DOS version
61	Open file
66	Move file pointer
63	Read file or device (Read 6800 bytes on handle 5)
66	Move file pointer
64	Write file or device (Write 6800 bytes on handle 5)
66	Move file pointer
64	Write file or device (Write 0 bytes on handle 5)
87	Get or set file date and time
62	Close file
48	Get DOS version
41	Parse filename
41	Parse filename
75	Execute program
9	Display string
76	Terminate with return code (Return code = '36')
64	Write file or device (Write 0 bytes on handle 1)

```
Syscall Op   |  Syscall Name
48    Get DOS version
61    Open file
66    Move file pointer
63    Read file or device (Read 2 bytes on
handle 5)
62    Close file
48    Get DOS version
61    Open file
66    Move file pointer
63    Read file or device (Read 6800 bytes on
handle 5)
66    Move file pointer
64    Write file or device (Write 6800 bytes on
handle 5)
66    Move file pointer
64    Write file or device (Write 0 bytes on
handle 5)
87    Get or set file date and time
62    Close file
48    Get DOS version
41    Parse filename
41    Parse filename
75    Execute program
9     Display string
76    Terminate with return code (Return code
= '36')
64    Write file or device (Write 0 bytes on
handle 1)
```

```
Syscall Op   |  Syscall Name
48    Get DOS version
61    Open file
66    Move file pointer
63    Read file or device (Read 2 bytes on
handle 5)
62    Close file
48    Get DOS version
61    Open file
66    Move file pointer
63    Read file or device (Read 6800 bytes on
handle 5)
66    Move file pointer
64    Write file or device (Write 6800 bytes on
handle 5)
66    Move file pointer
64    Write file or device (Write 0 bytes on
handle 5)
87    Get or set file date and time
62    Close file
48    Get DOS version
41    Parse filename
41    Parse filename
75    Execute program
9     Display string
76    Terminate with return code (Return code
= '36')
64    Write file or device (Write 0 bytes on
handle 1)
```

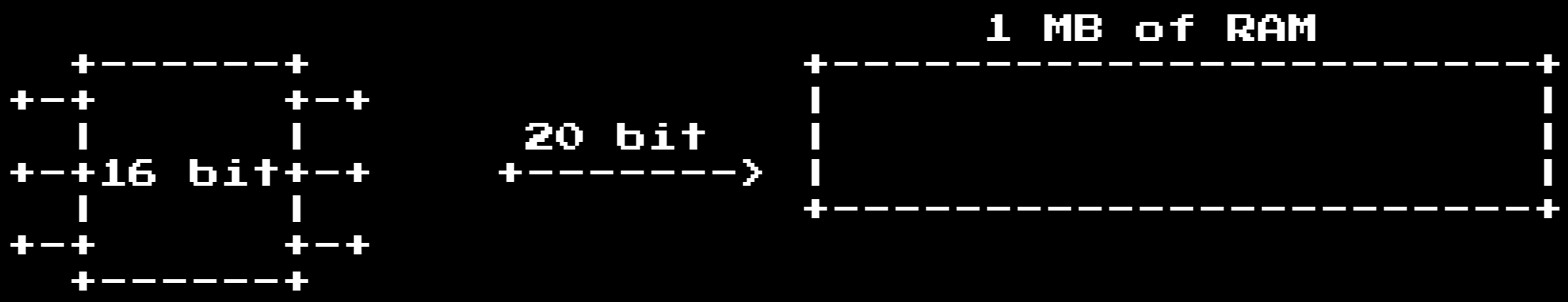
```
Syscall Op  |Syscall Name
48  Get DOS version
61  Open file
66  Move file pointer
63  Read file or device (Read 2 bytes on handle 5)
62  Close file
48  Get DOS version
61  Open file
66  Move file pointer
63  Read file or device (Read 6800 bytes on handle 5)
66  Move file pointer
64  Write file or device (Write 6800 bytes on handle 5)
66  Move file pointer
64  Write file or device (Write 0 bytes on handle 5)
87  Get or set file date and time
62  Close file
48  Get DOS version
41  Parse filename
41  Parse filename
75  Execute program
9   Display string
76  Terminate with return code (Return code = '36')
64  Write file or device (Write 0 bytes on handle 1)
```

```
AX: 0000 BX: DE00 CX: 929F DX: 0116
SI: F918 DI: 0116 SP: 0A66 BP: 0A72
CS: F000 DS: 0116 ES: 0040 SS: 0116

IP: 92BD EIP:000092BD
CS:IP: F000:92BD (0xF92BD)
SS:SP: 0116:0A66 (0x01BC6)
SS:BP: 0116:0A72 (0x01BD2)
```

```
AH = 09
DS:DX = pointer to string ending in "$"
```

Memory Layouts



2^16 = 64KB

2^20 = 1MB

Memory Layouts

16 BIT CPU

Segment:

CS, DS, SS, ES

"General" Use:

AX, BX, CX, DX, SP, BP, SI, DI

2^16 = 65536

2^16 = 65536

2^20 = 1048576

2^16 = 65536

2^20 = 1048576

1048576 = 1MiB

Memory Layouts

16 BIT CPU

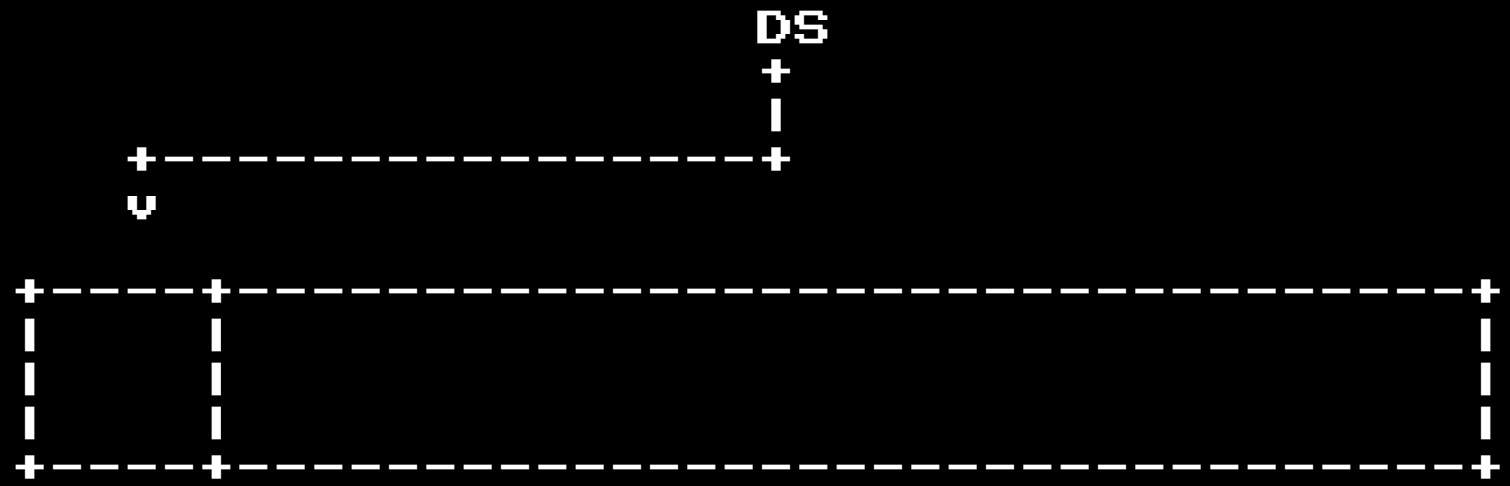
Segment:

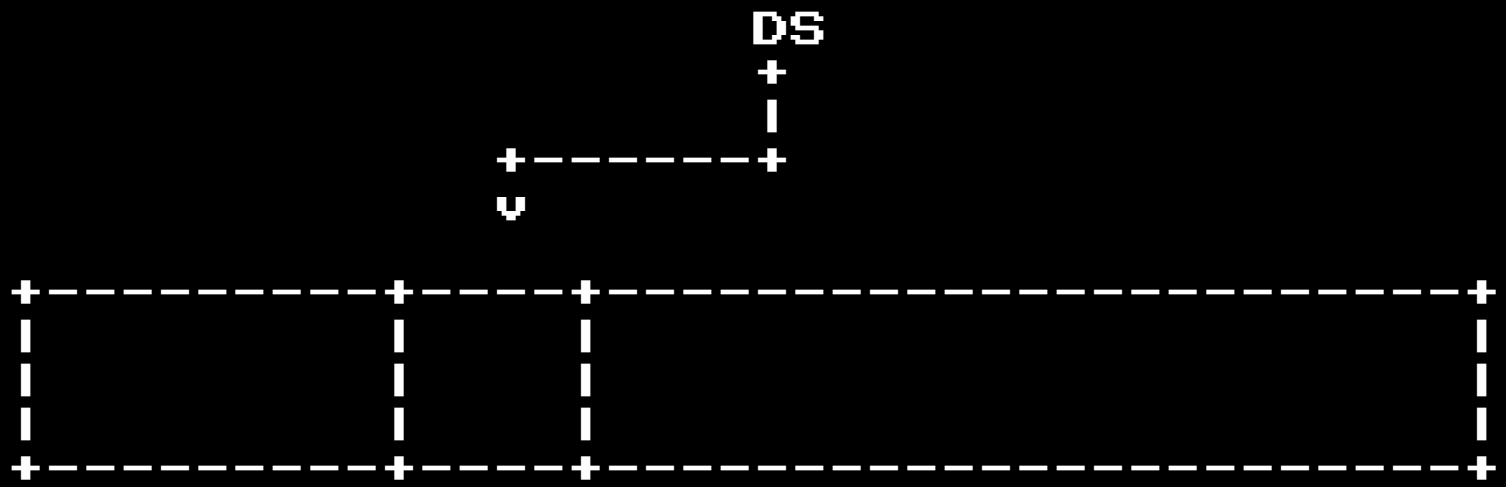
CS, DS, SS, ES

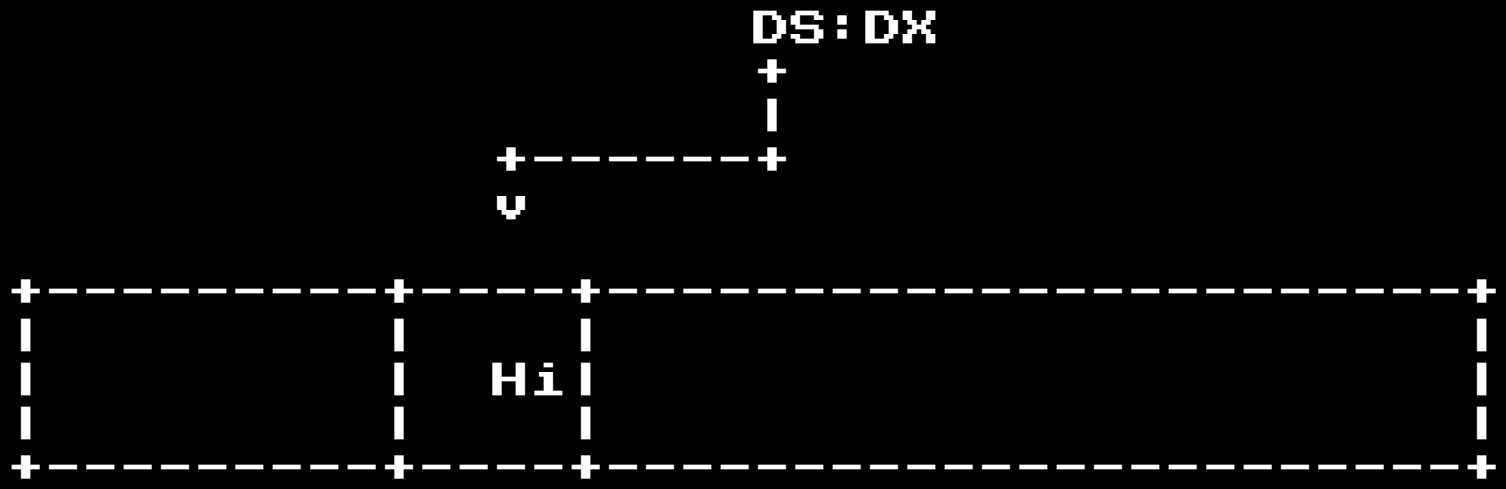
"General" Use:

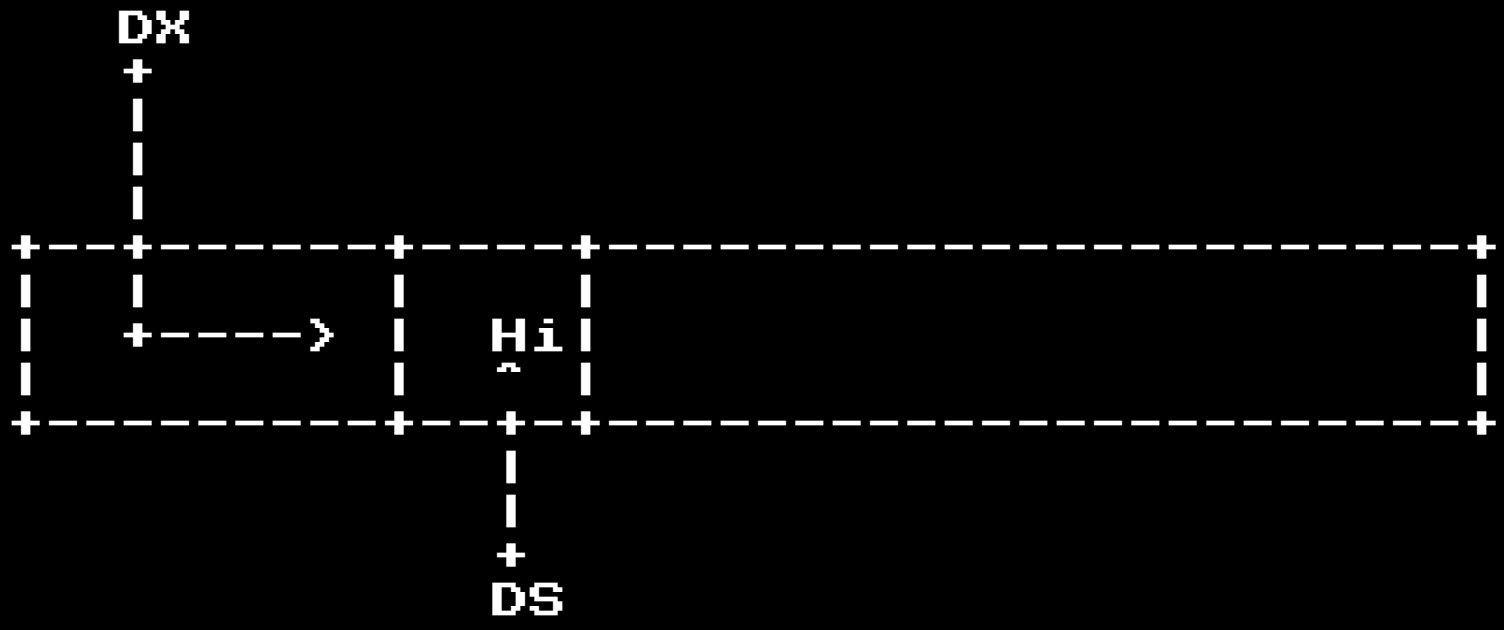
AX, BX, CX, DX, SP, BP, SI, DI

A quick crash course in segment registers


DS


DS


DS:DX
Hi


DX
Hi
DS


DX
DX: 6
DS: 100
ADDR: (100*16)+6
Hi
DS

16 BIT CPU

Segment:

CS, DS, SS, ES

"General" Use:

AX, BX, CX, DX, SP, BP, SI, DI

32 BIT CPU

Segment:

CS, DS, SS, ES

"General" Use:

EAX, EBX, ECX, EDX,
ESP, EBP, ESI, EDI

\+ Some new ones

64 BIT CPU

Segment:

CS, DS, SS, ES

"General" Use:

RAX, RBX, RCX, RDX,
RSP, RBP, RSI, RDI

\+ Loads of new ones

Tracing checklist

* Breakpoint on Int 21 handler

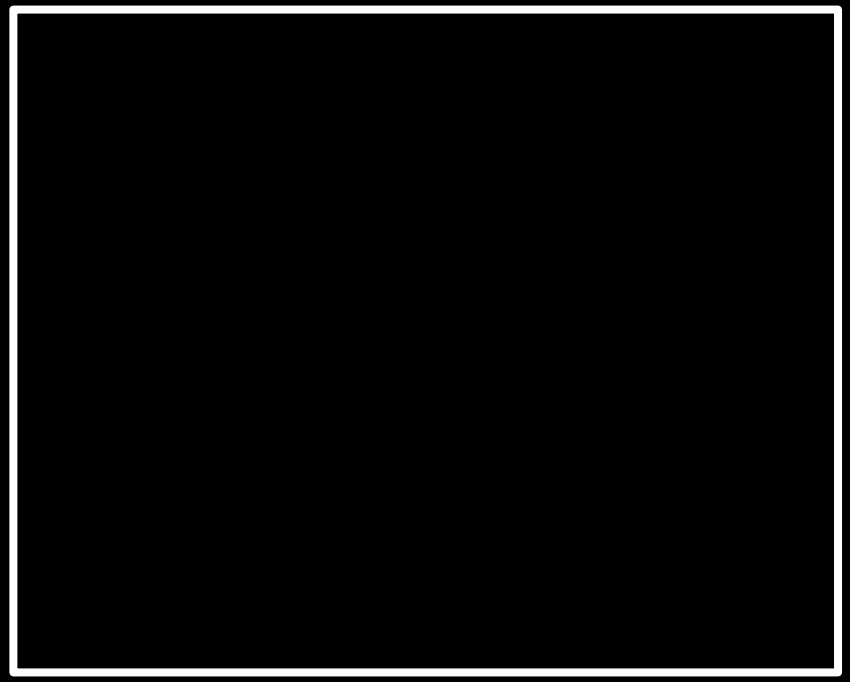
Tracing checklist

* Breakpoint on Int 21 handler
* Save registers

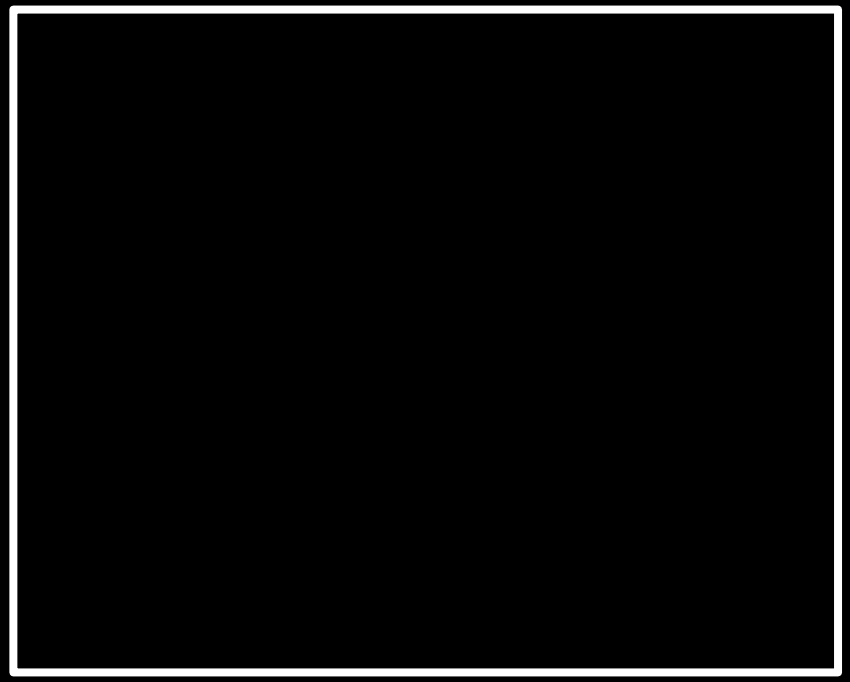
Tracing checklist

* Breakpoint on Int 21 handler
* Save registers
* Save 100 bytes from (DS * 16 + DX)
* Also record the screen for quick analysis

Drumroll...



Syscall Op	Syscall Name
48	Get DOS version
42	Get date
255	UNKNOWN!
73	Release memory
72	Allocate memory
74	Reallocate memory
74	Reallocate memory
76	Terminate with return code



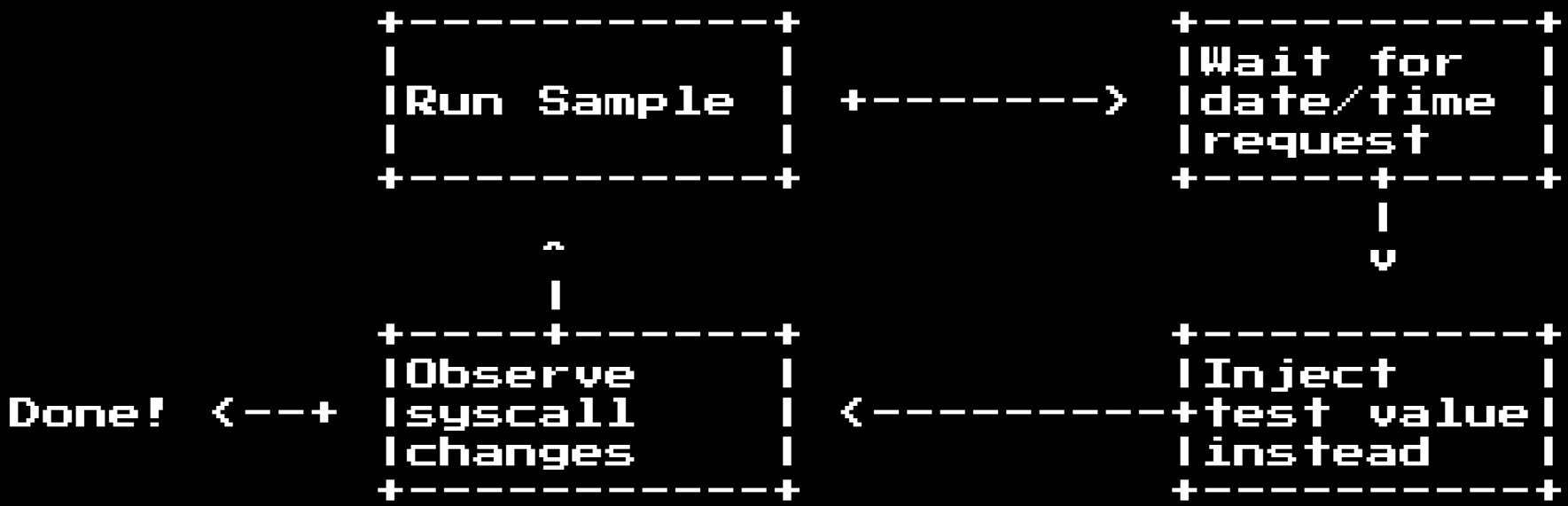
Syscall Op	Syscall Name
48	Get DOS version
42	Get date
255	UNKNOWN!
73	Release memory
72	Allocate memory
74	Reallocate memory
74	Reallocate memory
76	Terminate with return code

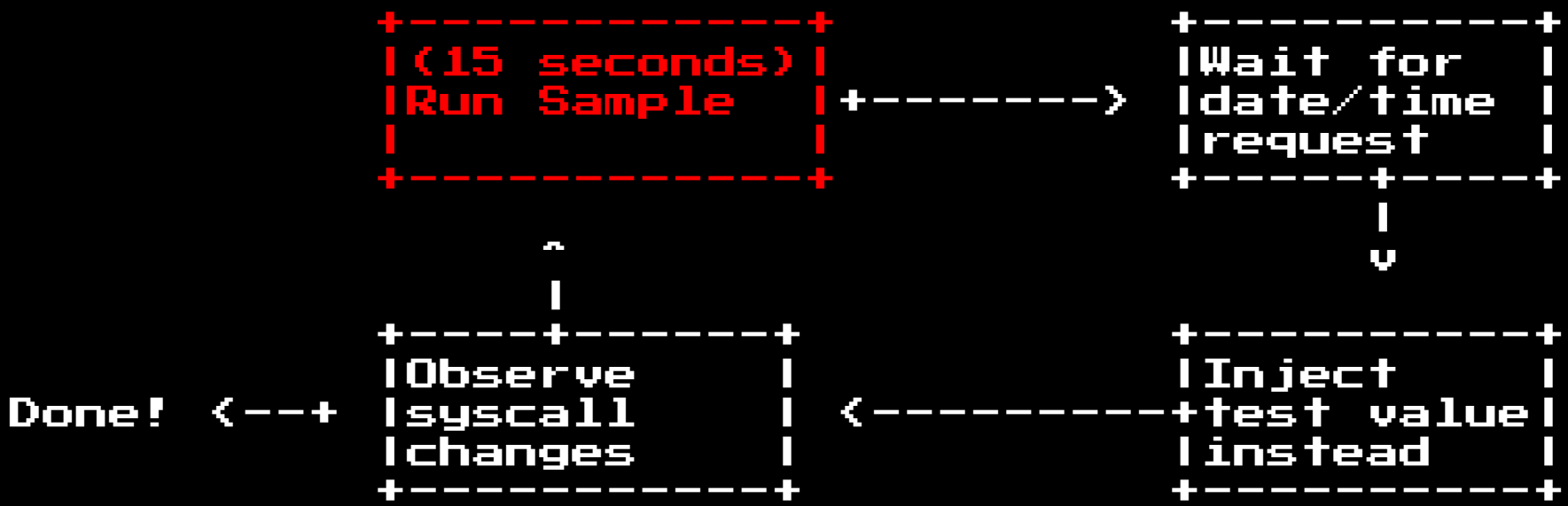
Int 21h
AH = 2A (Get Date)

on return:
AL = day of the week (0=Sunday)
CX = year (1980-2099)
DH = month (1-12)
DL = day (1-31)

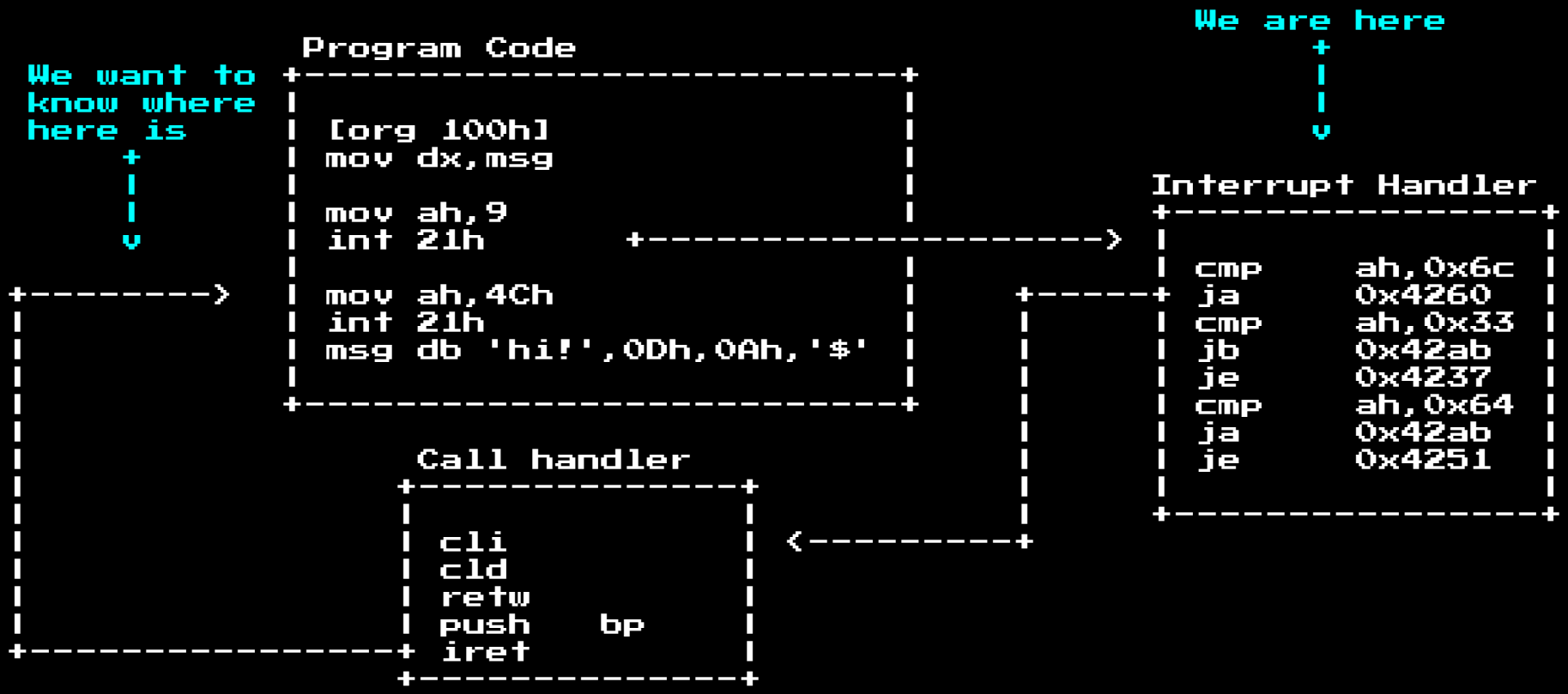
Int 21h
AH = 2C (Get Time)

on return:
CH = hour (0-23)
CL = minutes (0-59)
DH = seconds (0-59)
DL = hundredths (0-99)


Run Sample
Wait for date/time request
Inject test value instead
Observe syscall changes
Done!


(15 seconds)
Run Sample
Wait for
date/time
request
Inject
test value
instead
Observe
syscall
changes
Done!

OR


We are here
Program Code
We want to know where here is
[org 100h]
mov dx,msg
mov ah,9
int 21h
mov ah,4Ch
int 21h
msg db 'hi!',0Dh,0Ah,'$'
Interrupt Handler
cmp ah,0x6c
ja 0x4260
cmp ah,0x33
jb 0x42ab
je 0x4237
cmp ah,0x64
ja 0x42ab
je 0x4251
Call handler
cli
cld
retw
push bp
iret


IP
CS
+[STACK]+
010A 1294 7246 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
Located at SS:SP

Tracing checklist

* Breakpoint on Int 21 handler
* Save registers
* Save 100 bytes from (DS * 16 + DX)
* Also record the screen for quick analysis
* Grab 4 bytes from SS:SP

Tracing checklist

* Breakpoint on Int 21 handler
* Save registers
* Save 100 bytes from (DS * 16 + DX)
* Also record the screen for quick analysis
* Grab 4 bytes from SS:SP
* Grab 100 bytes from the return address

```
0x12b69:   cmp   dl, 0x1e
0x12b6c:   je 0x12b70
0x12b6e:   jmp   0x12bb9
0x12b70:   mov   ah, 0x4e
0x12b72:   mov   cx, 7
0x12b75:   lea   dx, word ptr [bp + 0x508]
0x12b79:   int   0x21
0x12b7b:   jae   0x12b7f
0x12b7d:   jmp   0x12b9c
0x12b7f:   mov   ax, 0x3d02
0x12b82:   lea   dx, word ptr [bp + 0x55e]
0x12b86:   int   0x21
0x12b88:   xchg  ax, bx
0x12b89:   mov   ah, 0x40
0x12b8b:   mov   cx, 0x71
0x12b8e:   lea   dx, word ptr [bp + 0x2b5]
0x12b92:   int   0x21
0x12b94:   mov   ah, 0x3e
0x12b96:   int   0x21
0x12b98:   mov   ah, 0x4f
```

```
0x12b69:   cmp  dl, 0x1e
0x12b6c:   je 0x12b70
0x12b6e:   jmp  0x12bb9
0x12b70:   mov  ah, 0x4e
0x12b72:   mov  cx, 7
0x12b75:   lea  dx, word ptr [bp + 0x508]
0x12b79:   int  0x21
0x12b7b:   jae  0x12b7f
0x12b7d:   jmp  0x12b9c
0x12b7f:   mov  ax, 0x3d02
0x12b82:   lea  dx, word ptr [bp + 0x55e]
0x12b86:   int  0x21
0x12b88:   xchg ax, bx
0x12b89:   mov  ah, 0x40
0x12b8b:   mov  cx, 0x71
0x12b8e:   lea  dx, word ptr [bp + 0x2b5]
0x12b92:   int  0x21
0x12b94:   mov  ah, 0x3e
0x12b96:   int  0x21
0x12b98:   mov  ah, 0x4f
```

Int 21h
AH = 2A (Get Date)

on return:
AL = day of the week (0=Sunday)
CX = year (1980-2099)
DH = month (1-12)
DL = day (1-31)

Int 21h
AH = 2C (Get Time)

on return:
CH = hour (0-23)
CL = minutes (0-59)
DH = seconds (0-59)
DL = hundredths (0-99)

```
Int 21h
AH = 2A (Get Date)


on return:
AL = day of the week (0=Sunday)
CX = year (1980-2099)
DH = month (1-12)
DL = day (1-31)


Int 21h
AH = 2C (Get Time)


on return:
CH = hour (0-23)
CL = minutes (0-59)
DH = seconds (0-59)
DL = hundredths (0-99)
```

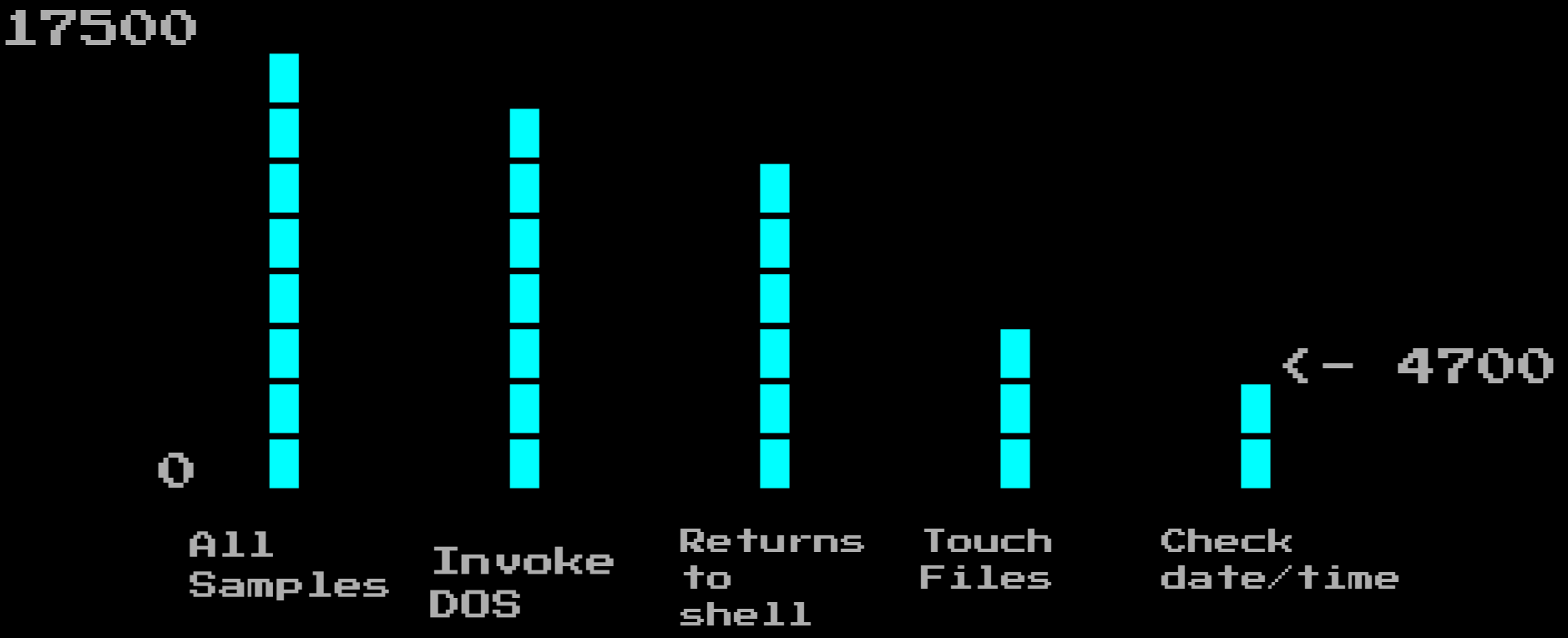
```
0x1e = 30
DL = Day of Month


0x12b69:   cmp   dl, 0x1e
0x12b6c:   je 0x12b70
0x12b6e:   jmp   0x12bb9
```

```
0x1e = 30
DL = Day of Month


0x12b69:   cmp  dl, 0x1e
0x12b6c:   je 0x12b70
0x12b6e:   jmp  0x12bb9
```

```
If ($dayOfMonth == 30) {
  Goto 0x12b70;
} else {
  Goto 0x12bb9;
}
```


17500
0
(- 4700
All Samples
Invoke DOS
Returns to shell
Touch Files
Check date/time

The world's worst x86 emulator

BenX86

* 16 bit only
* Any pointer memory access ends emulation
* Fake stack, push = nop, pop = end of emulation
* 80~ opcodes implemented (Most of them are jumps)
* Logs every opcode that is ran
* Can be run with just a x86 code snippet and a register snapshot

BenX86

* All days from 1980 to 2005 (9125 days) can be tested in ~100ms

* Most programs have 3~ code paths based on dates

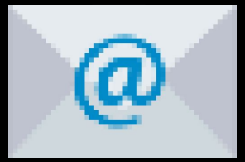
* That yields us...

17k
samples

10k date/time
variations

Thank you

Questions / Corrections?



ben@benjojo.co.uk

@benjojo12 / Find me at the Tea House

I'm on the hunt of a europe based job from April 2019