# ARHITECTURA SISTEMELOR DE CALCUL - CURS 0x0C

## BOOTLOADER

Cristian Rusu

# DATA TRECUTĂ

- fișiere binare ELF

- sistemul de operare: procese și spațiul memoriei

# CUPRINS

- secvența de boot (detaliată)
- bootloader simplu
- astăzi folosim Netwide Assembly (NASM) și Windows

# SECVENȚA DE BOOT

- **la pornirea calculatorului este activat BIOS-ul**

- **BIOS-ul este în RAM:**
  - realizează Power-On Self Test (POST procedure)
  - încarcă bootloader-ul
  - scopul este găsirea sistemului de operare și rularea sa
  - OS-ul este căutat pe HDD/SSD/CD-ROM/USB/floppy

- **unde este bootloader-ul?**
  - primul sector (primii 512 bytes) de pe dispozitiv
  - de unde știm că e bootloader? magic number: 0xAA55

- **bootloader-ul găsit este încărcat în memorie la 0x7C00**

# SECVENȚA DE BOOT

- **unde este bootloader-ul?**
  - primul sector (primii 512 bytes) de pe dispozitiv
  - de unde știm că e bootloader? magic number: 0xAA55

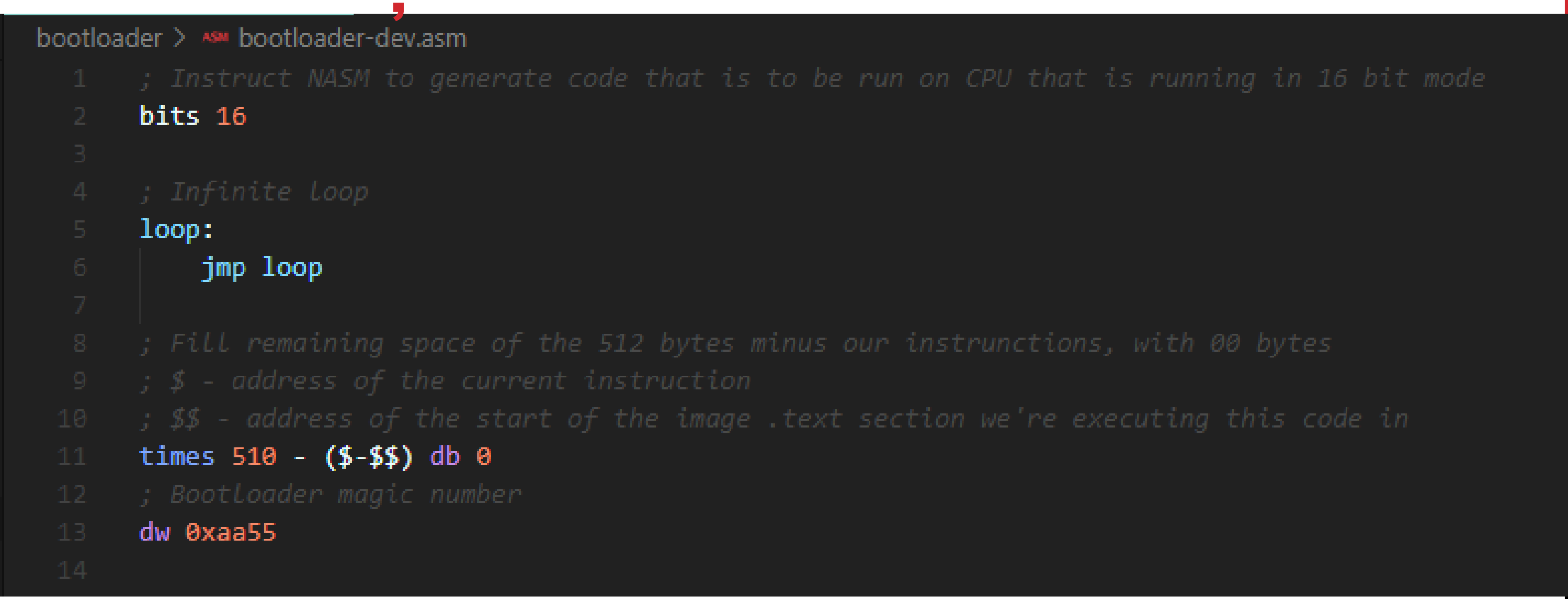
- **bootloader-ul găsit este încărcat în memorie la 0x7C00**

- **pentru că ”primul bootloader” este limitat la 512 bytes, acesta încarcă defapt încă un bootloader care nu mai are limitări**

- **pe Windows, bootloader-ul este la Windows\System32\ntoskrnl.exe**

- **în tot acest timp, procesorul este în modul de lucru pe 16 biți**

# SECVENȚA DE BOOT

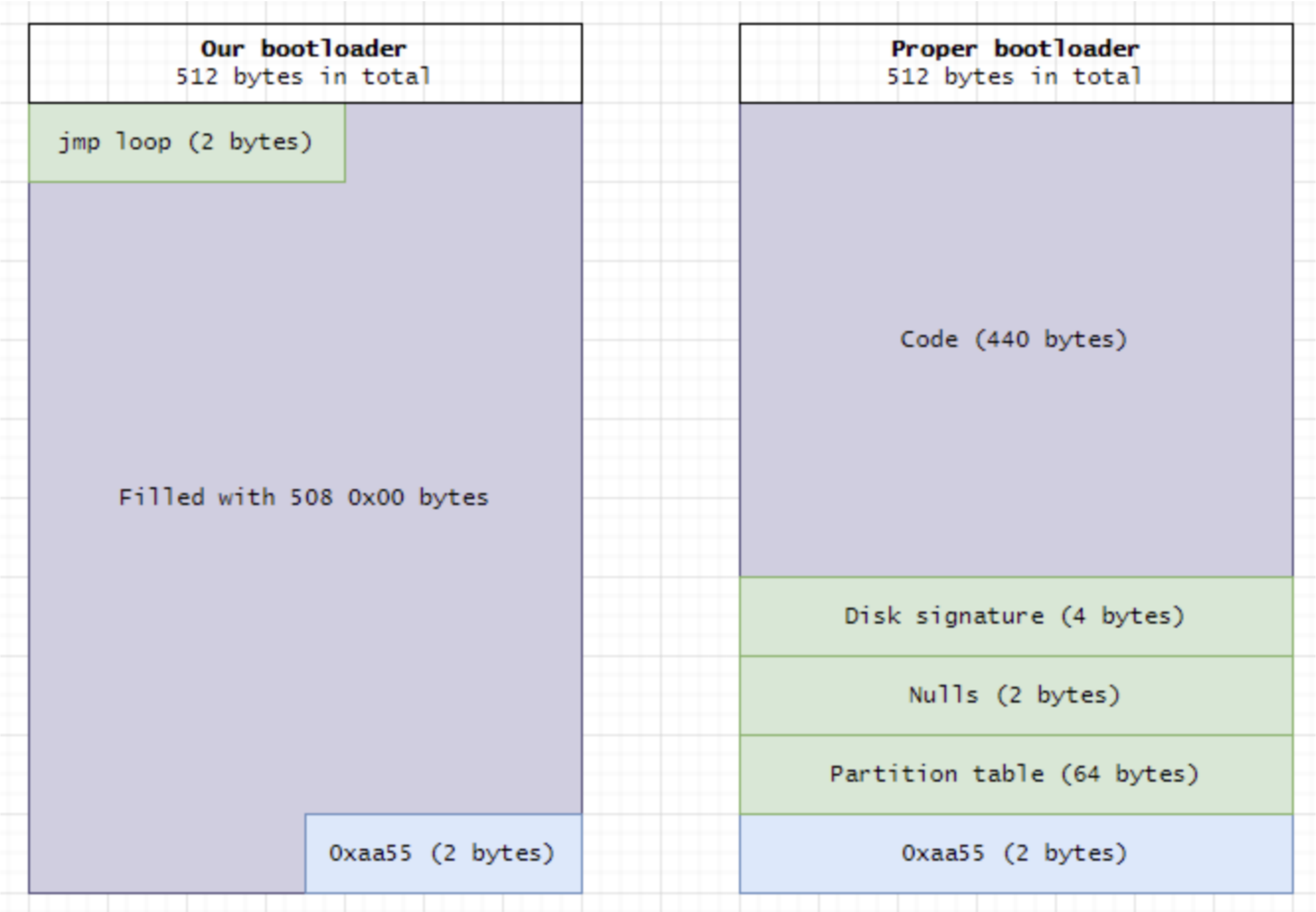


```
; Instruct NASM to generate code that is to be run on CPU that is running in 16 bit mode
bits 16

; Infinite loop
loop:
    jmp loop

; Fill remaining space of the 512 bytes minus our instrunctions, with 00 bytes
; $ - address of the current instruction
; $$ - address of the start of the image .text section we're executing this code in
times 510 - ($-$$) db 0
; Bootloader magic number
dw 0xaa55
```

- **CPU funcționeaza pe 16 biți**
- **$ - adresa instrucțiunii actuale**
- **$$ - adresa secțiunii .text**

```
nasm -f bin bootloader-dev.asm -o bootloader.bin
```

# SECVENȚA DE BOOT



**Our bootloader**
512 bytes in total

jmp loop (2 bytes)

Filled with 508 0x00 bytes

0xaa55 (2 bytes)

**Proper bootloader**
512 bytes in total

Code (440 bytes)

Disk signature (4 bytes)

Nulls (2 bytes)

Partition table (64 bytes)

0xaa55 (2 bytes)

```
nasm -f bin bootloader-dev.asm -o bootloader.bin
```

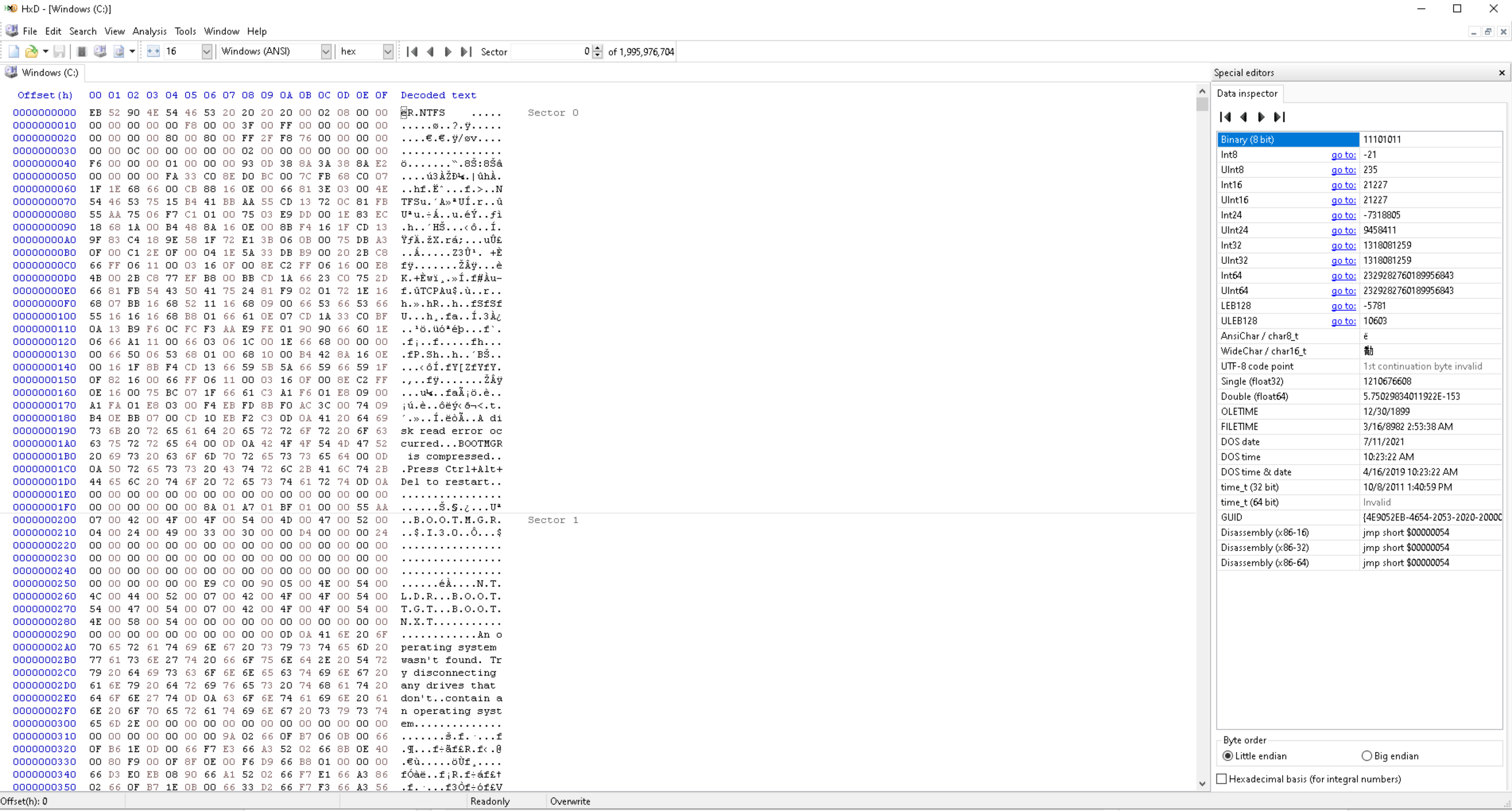
# HXD

- **vom folosi tool-ul HxD pentru a verifica conținutul HD**

- **HxD este un tool pentru e vizualiza/edita:**
  - HD/SSD
  - fișiere
  - procese

https://mh-nexus.de/en/hxd/

# HXD



16 | Windows (ANSI) | hex | Sector 0 of 1,995,976,704

```
 Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

0000000000  EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëR.NTFS    .....     Sector 0
0000000010  00 00 00 00 00 F8 00 00 3F 00 FF 00 00 00 00 00  .....ø..?.ÿ.....
0000000020  00 00 00 00 80 00 80 00 FF 2F F8 76 00 00 00 00  ....€.€.ÿ/øv....
0000000030  00 00 0C 00 00 00 00 00 02 00 00 00 00 00 00 00  ................
0000000040  F6 00 00 00 01 00 00 00 93 0D 38 8A 3A 38 8A E2  ö.......".8Š:8Šâ
0000000050  00 00 00 00 FA 33 C0 8E D0 BC 00 7C FB 68 C0 07  ....ú3ÀŽÐ¼.|ûhÀ.
0000000060  1F 1E 68 66 00 CB 88 16 0E 00 66 81 3E 03 00 4E  ..hf.Ëˆ...f.>..N
0000000070  54 46 53 75 15 B4 41 BB AA 55 CD 13 72 0C 81 FB  TFSu.´A»ªUÍ.r..û
0000000080  55 AA 75 06 F7 C1 01 00 75 03 E9 DD 00 1E 83 EC  Uªu.÷Á..u.éÝ..ƒì
0000000090  18 68 1A 00 B4 48 8A 16 0E 00 8B F4 16 1F CD 13  .h..´HŠ...‹ô..Í.
00000000A0  9F 83 C4 18 9E 58 1F 72 E1 3B 06 0B 00 75 DB A3  ŸƒÄ.žX.rá;...uÛ£
00000000B0  0F 00 C1 2E 0F 00 04 1E 5A 33 DB B9 00 20 2B C8  ..Á.....Z3Û¹. +È
00000000C0  66 FF 06 11 00 03 16 0F 00 8E C2 FF 06 16 00 E8  fÿ.......ŽÂÿ...è
00000000D0  4B 00 2B C8 77 EF B8 00 BB CD 1A 66 23 C0 75 2D  K.+Èwï¸.»Í.f#Àu-
00000000E0  66 81 FB 54 43 50 41 75 24 81 F9 02 01 72 1E 16  f.ûTCPAu$.ù..r..
00000000F0  68 07 BB 16 68 52 11 16 68 09 00 66 53 66 53 66  h.».hR..h..fSfSf
0000000100  55 16 16 16 68 B8 01 66 61 0E 07 CD 1A 33 C0 BF  U...h¸.fa..Í.3À¿
0000000110  0A 13 B9 F6 0C FC F3 AA E9 FE 01 90 90 66 60 1E  ..¹ö.üóªéþ...f`.
0000000120  06 66 A1 11 00 66 03 06 1C 00 1E 66 68 00 00 00  .f¡..f.....fh...
0000000130  00 66 50 06 53 68 01 00 68 10 00 B4 42 8A 16 0E  .fP.Sh..h..´BŠ..
0000000140  00 16 1F 8B F4 CD 13 66 59 5B 5A 66 59 66 59 1F  ...‹ôÍ.fY[ZfYfY.
0000000150  0F 82 16 00 66 FF 06 11 00 03 16 0F 00 8E C2 FF  .‚..fÿ.......ŽÂÿ
0000000160  0E 16 00 75 BC 07 1F 66 61 C3 A1 F6 01 E8 09 00  ...u¼..faÃ¡ö.è..
0000000170  A1 FA 01 E8 03 00 F4 EB FD 8B F0 AC 3C 00 74 09  ¡ú.è..ôëý‹ð¬<.t.
0000000180  B4 0E BB 07 00 CD 10 EB F2 C3 0D 0A 41 20 64 69  ´.»..Í.ëòÃ..A di
0000000190  73 6B 20 72 65 61 64 20 65 72 72 6F 72 20 6F 63  sk read error oc
00000001A0  63 75 72 72 65 64 00 0D 0A 42 4F 4F 54 4D 47 52  curred...BOOTMGR
00000001B0  20 69 73 20 63 6F 6D 70 72 65 73 73 65 64 00 0D   is compressed..
00000001C0  0A 50 72 65 73 73 20 43 74 72 6C 2B 41 6C 74 2B  .Press Ctrl+Alt+
00000001D0  44 65 6C 20 74 6F 20 72 65 73 74 61 72 74 0D 0A  Del to restart..
00000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000001F0  00 00 00 00 00 00 8A 01 A7 01 BF 01 00 00 55 AA  ......Š.§.¿...Uª
0000000200  07 00 42 00 4F 00 4F 00 54 00 4D 00 47 00 52 00  ..B.O.O.T.M.G.R.     Sector 1
0000000210  04 00 24 00 49 00 33 00 30 00 00 D4 00 00 00 24  ..$.I.3.0..Ô...$
0000000220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000000230  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000000240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000000250  00 00 00 00 00 00 E9 C0 00 90 05 00 4E 00 54 00  ......éÀ....N.T.
0000000260  4C 00 44 00 52 00 07 00 42 00 4F 00 4F 00 54 00  L.D.R...B.O.O.T.
0000000270  54 00 47 00 54 00 07 00 42 00 4F 00 4F 00 54 00  T.G.T...B.O.O.T.
0000000280  4E 00 58 00 54 00 00 00 00 00 00 00 00 00 00 00  N.X.T...........
0000000290  00 00 00 00 00 00 00 00 00 00 0D 0A 41 6E 20 6F  ............An o
00000002A0  70 65 72 61 74 69 6E 67 20 73 79 73 74 65 6D 20  perating system
00000002B0  77 61 73 6E 27 74 20 66 6F 75 6E 64 2E 20 54 72  wasn't found. Tr
00000002C0  79 20 64 69 73 63 6F 6E 6E 65 63 74 69 6E 67 20  y disconnecting
00000002D0  61 6E 79 20 64 72 69 76 65 73 20 74 68 61 74 20  any drives that
00000002E0  64 6F 6E 27 74 0D 0A 63 6F 6E 74 61 69 6E 20 61  don't..contain a
00000002F0  6E 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  n operating syst
0000000300  65 6D 2E 00 00 00 00 00 00 00 00 00 00 00 00 00  em..............
0000000310  00 00 00 00 00 00 00 9A 02 66 0F B7 06 0B 00 66  .......š.f.·...f
0000000320  0F B6 1E 0D 00 66 F7 E3 66 A3 52 02 66 8B 0E 40  .¶...f÷ãf£R.f‹.@
0000000330  00 80 F9 00 0F 8F 0E 00 F6 D9 66 B8 01 00 00 00  .€ù.....öÙf¸....
0000000340  66 D3 E0 EB 08 90 66 A1 52 02 66 F7 E1 66 A3 86  fÓàë..f¡R.f÷áf£†
0000000350  02 66 0F B7 1E 0B 00 66 33 D2 66 F7 F3 66 A3 56  .f.·...f3Òf÷óf£V
```

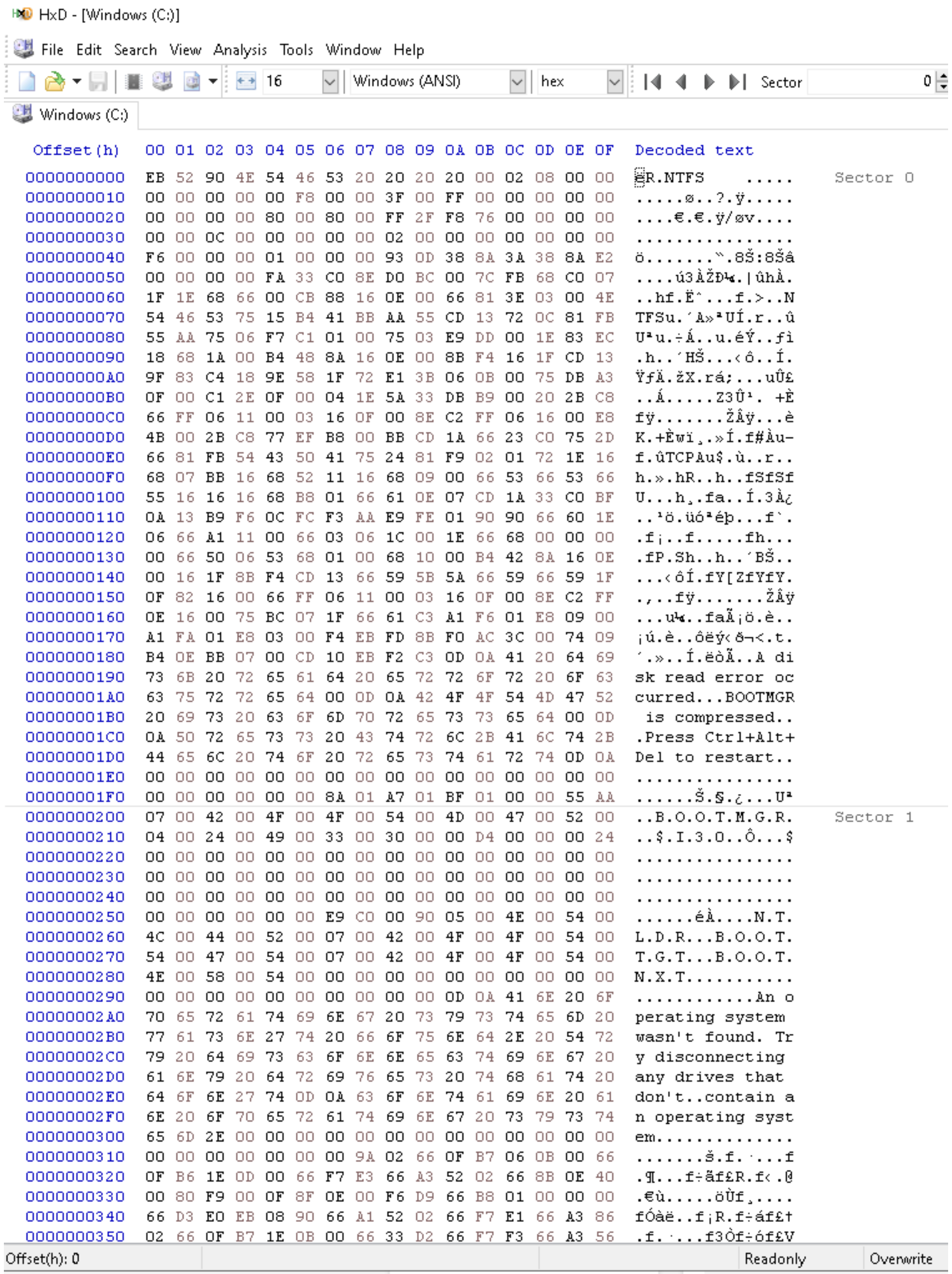
Offset(h): 0 | Readonly | Overwrite

|  |  |  |
|---|---|---|
| Binary (8 bit) |  | 11101011 |
| Int8 | go to: | -21 |
| UInt8 | go to: | 235 |
| Int16 | go to: | 21227 |
| UInt16 | go to: | 21227 |
| Int24 | go to: | -7318805 |
| UInt24 | go to: | 9458411 |
| Int32 | go to: | 1318081259 |
| UInt32 | go to: | 1318081259 |
| Int64 | go to: | 2329282760189956843 |
| UInt64 | go to: | 2329282760189956843 |
| LEB128 | go to: | -5781 |
| ULEB128 | go to: | 10603 |
| AnsiChar / char8_t |  | ë |
| WideChar / char16_t |  | 勫 |
| UTF-8 code point |  | 1st continuation byte invalid |
| Single (float32) |  | 1210676608 |
| Double (float64) |  | 5.75029834011922E-153 |
| OLETIME |  | 12/30/1899 |
| FILETIME |  | 3/16/8982 2:53:38 AM |
| DOS date |  | 7/11/2021 |
| DOS time |  | 10:23:22 AM |
| DOS time & date |  | 4/16/2019 10:23:22 AM |
| time_t (32 bit) |  | 10/8/2011 1:40:59 PM |
| time_t (64 bit) |  | Invalid |
| GUID |  | {4E9052EB-4654-2053-2020-2000C |
| Disassembly (x86-16) |  | jmp short $00000054 |
| Disassembly (x86-32) |  | jmp short $00000054 |
| Disassembly (x86-64) |  | jmp short $00000054 |

Byte order
● Little endian ○ Big endian

□ Hexadecimal basis (for integral numbers)

# HXD



```
 Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

0000000000  EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëR.NTFS    .....    Sector 0
0000000010  00 00 00 00 00 F8 00 00 3F 00 FF 00 00 00 00 00  .....ø..?.ÿ.....
0000000020  00 00 00 00 80 00 80 00 FF 2F F8 76 00 00 00 00  ....€.€.ÿ/øv....
0000000030  00 00 0C 00 00 00 00 00 02 00 00 00 00 00 00 00  ................
0000000040  F6 00 00 00 01 00 00 00 93 0D 38 8A 3A 38 8A E2  ö.......“.8Š:8Šâ
0000000050  00 00 00 00 FA 33 C0 8E D0 BC 00 7C FB 68 C0 07  ....ú3ÀŽÐ¼.|ûhÀ.
0000000060  1F 1E 68 66 00 CB 88 16 0E 00 66 81 3E 03 00 4E  ..hf.Ëˆ...f.>..N
0000000070  54 46 53 75 15 B4 41 BB AA 55 CD 13 72 0C 81 FB  TFSu.´A»ªUÍ.r..û
0000000080  55 AA 75 06 F7 C1 01 00 75 03 E9 DD 00 1E 83 EC  Uªu.÷Á..u.éÝ..ƒì
0000000090  18 68 1A 00 B4 48 8A 16 0E 00 8B F4 16 1F CD 13  .h..´HŠ...‹ô..Í.
00000000A0  9F 83 C4 18 9E 58 1F 72 E1 3B 06 0B 00 75 DB A3  ŸƒÄ.žX.rá;...uÛ£
00000000B0  0F 00 C1 2E 0F 00 04 1E 5A 33 DB B9 00 20 2B C8  ..Á.....Z3Û¹. +È
00000000C0  66 FF 06 11 00 03 16 0F 00 8E C2 FF 06 16 00 E8  fÿ.......ŽÂÿ...è
00000000D0  4B 00 2B C8 77 EF B8 00 BB CD 1A 66 23 C0 75 2D  K.+Èwï¸.»Í.f#Àu-
00000000E0  66 81 FB 54 43 50 41 75 24 81 F9 02 01 72 1E 16  f.ûTCPAu$.ù..r..
00000000F0  68 07 BB 16 68 52 11 16 68 09 00 66 53 66 53 66  h.».hR..h..fSfSf
0000000100  55 16 16 16 68 B8 01 66 61 0E 07 CD 1A 33 C0 BF  U...h¸.fa..Í.3À¿
0000000110  0A 13 B9 F6 0C FC F3 AA E9 FE 01 90 90 66 60 1E  ..¹ö.üóªéþ...f`.
0000000120  06 66 A1 11 00 66 03 06 1C 00 1E 66 68 00 00 00  .f¡..f.....fh...
0000000130  00 66 50 06 53 68 01 00 68 10 00 B4 42 8A 16 0E  .fP.Sh..h..´BŠ..
0000000140  00 16 1F 8B F4 CD 13 66 59 5B 5A 66 59 66 59 1F  ...‹ôÍ.fY[ZfYfY.
0000000150  0F 82 16 00 66 FF 06 11 00 03 16 0F 00 8E C2 FF  .‚..fÿ.......ŽÂÿ
0000000160  0E 16 00 75 BC 07 1F 66 61 C3 A1 F6 01 E8 09 00  ...u¼..faÃ¡ö.è..
0000000170  A1 FA 01 E8 03 00 F4 EB FD 8B F0 AC 3C 00 74 09  ¡ú.è..ôëý‹ð¬<.t.
0000000180  B4 0E BB 07 00 CD 10 EB F2 C3 0D 0A 41 20 64 69  ´.»..Í.ëòÃ..A di
0000000190  73 6B 20 72 65 61 64 20 65 72 72 6F 72 20 6F 63  sk read error oc
00000001A0  63 75 72 72 65 64 00 0D 0A 42 4F 4F 54 4D 47 52  curred...BOOTMGR
00000001B0  20 69 73 20 63 6F 6D 70 72 65 73 73 65 64 00 0D   is compressed..
00000001C0  0A 50 72 65 73 73 20 43 74 72 6C 2B 41 6C 74 2B  .Press Ctrl+Alt+
00000001D0  44 65 6C 20 74 6F 20 72 65 73 74 61 72 74 0D 0A  Del to restart..
00000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000001F0  00 00 00 00 00 00 8A 01 A7 01 BF 01 00 00 55 AA  ......Š.§.¿...Uª
0000000200  07 00 42 00 4F 00 4F 00 54 00 4D 00 47 00 52 00  ..B.O.O.T.M.G.R.    Sector 1
0000000210  04 00 24 00 49 00 33 00 30 00 00 D4 00 00 00 24  ..$.I.3.0..Ô...$
0000000220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000000230  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000000240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000000250  00 00 00 00 00 00 E9 C0 00 90 05 00 4E 00 54 00  ......éÀ....N.T.
0000000260  4C 00 44 00 52 00 07 00 42 00 4F 00 4F 00 54 00  L.D.R...B.O.O.T.
0000000270  54 00 47 00 54 00 07 00 42 00 4F 00 4F 00 54 00  T.G.T...B.O.O.T.
0000000280  4E 00 58 00 54 00 00 00 00 00 00 00 00 00 00 00  N.X.T...........
0000000290  00 00 00 00 00 00 00 00 00 00 0D 0A 41 6E 20 6F  ............An o
00000002A0  70 65 72 61 74 69 6E 67 20 73 79 73 74 65 6D 20  perating system
00000002B0  77 61 73 6E 27 74 20 66 6F 75 6E 64 2E 20 54 72  wasn't found. Tr
00000002C0  79 20 64 69 73 63 6F 6E 6E 65 63 74 69 6E 67 20  y disconnecting
00000002D0  61 6E 79 20 64 72 69 76 65 73 20 74 68 61 74 20  any drives that
00000002E0  64 6F 6E 27 74 0D 0A 63 6F 6E 74 61 69 6E 20 61  don't..contain a
00000002F0  6E 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  n operating syst
0000000300  65 6D 2E 00 00 00 00 00 00 00 00 00 00 00 00 00  em..............
0000000310  00 00 00 00 00 00 00 9A 02 66 0F B7 06 0B 00 66  .......š.f.·...f
0000000320  0F B6 1E 0D 00 66 F7 E3 66 A3 52 02 66 8B 0E 40  .¶...f÷ãf£R.f‹.@
0000000330  00 80 F9 00 0F 8F 0E 00 F6 D9 66 B8 01 00 00 00  .€ù.....öÙf¸....
0000000340  66 D3 E0 EB 08 90 66 A1 52 02 66 F7 E1 66 A3 86  fÓàë..f¡R.f÷áf£†
0000000350  02 66 0F B7 1E 0B 00 66 33 D2 66 F7 F3 66 A3 56  .f.·...f3Òf÷óf£V
```

Offset(h): 0    Readonly    Overwrite

# QEMU

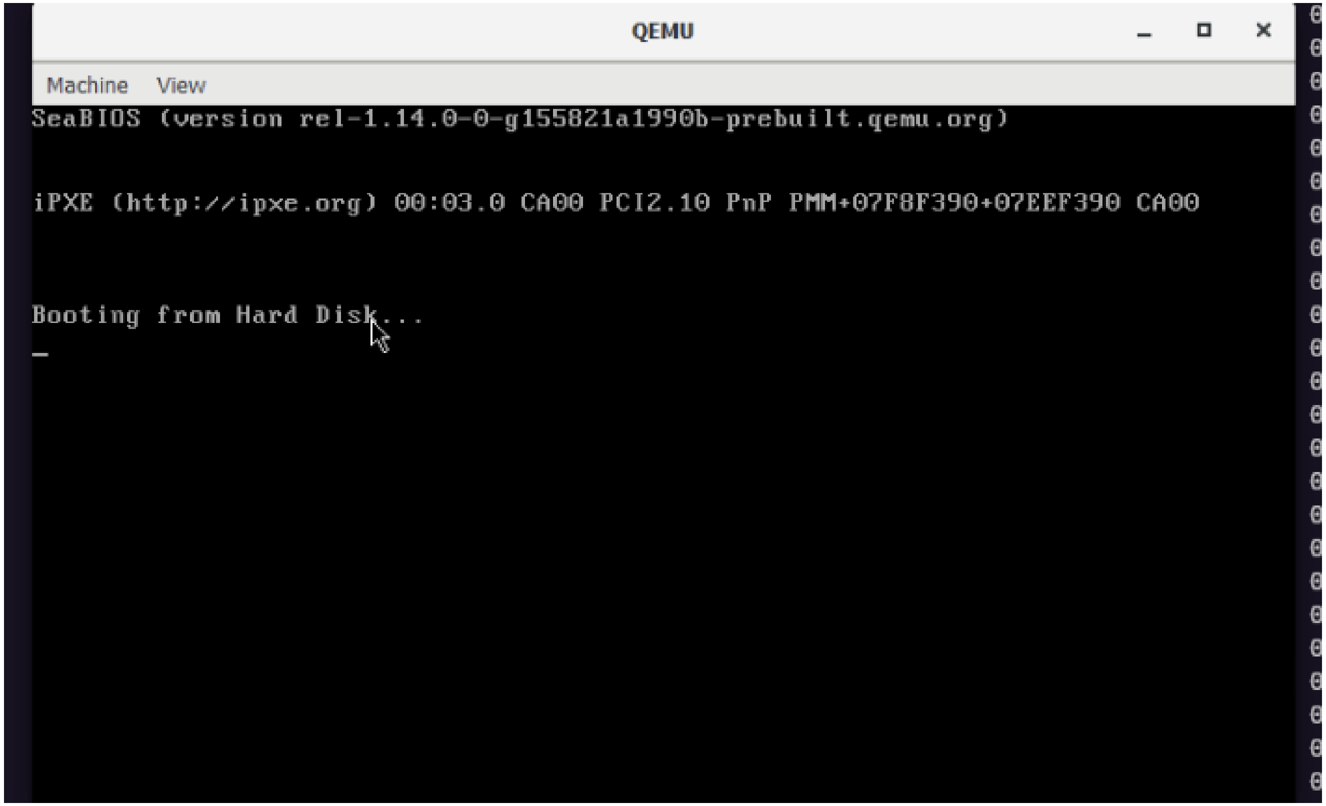
- **Quick Emulator (QEMU)**

- **emulator open-source**
  - emuleaza un procesor (și periferice etc.)
  - folosește traducere binară dinamică (dynamic binary translation)
  - putem testa secvența de boot

https://www.qemu.org/

# QEMU

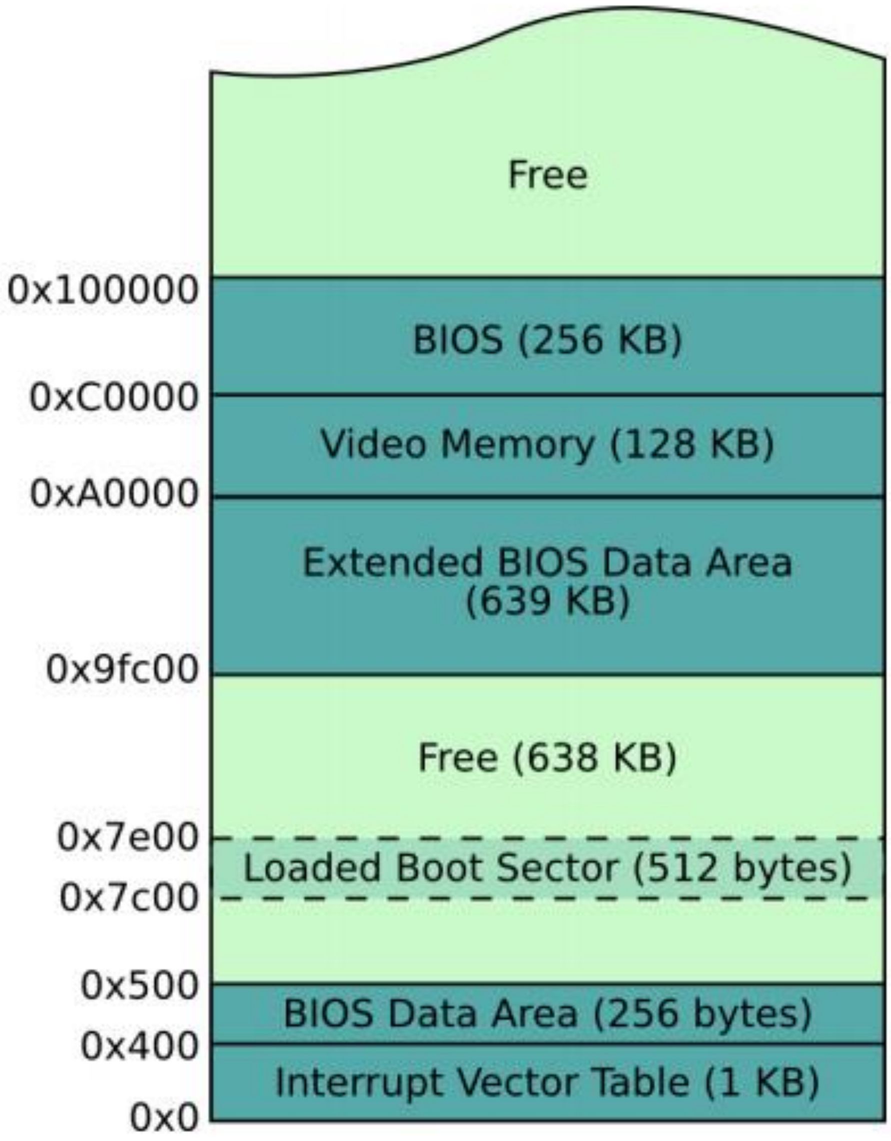
- rulăm programul nostru anterior
- qemu-system-x86_64.exe bootloader.bin



```
SeaBIOS (version rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8F390+07EEF390 CA00



Booting from Hard Disk...
_
```

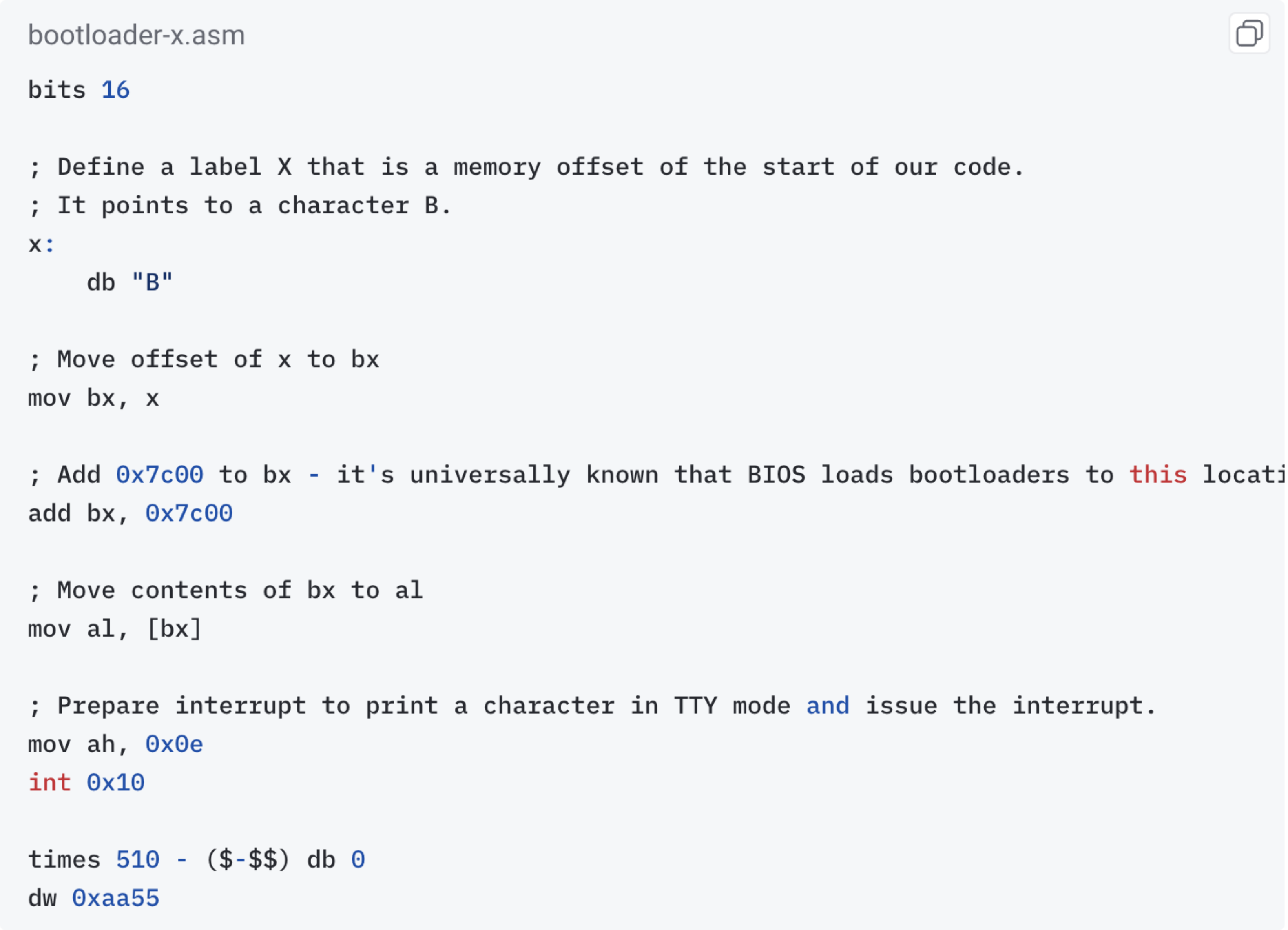
https://www.qemu.org/

# MEMORIA LA BOOT


Free
0x100000
BIOS (256 KB)
0xC0000
Video Memory (128 KB)
0xA0000
Extended BIOS Data Area (639 KB)
0x9fc00
Free (638 KB)
0x7e00
Loaded Boot Sector (512 bytes)
0x7c00
0x500
BIOS Data Area (256 bytes)
0x400
Interrupt Vector Table (1 KB)
0x0

https://www.cs.bham.ac.uk/~exr/lectures/opsys/10_11/lectures/os-dev.pdf

# MEMORIA LA BOOT

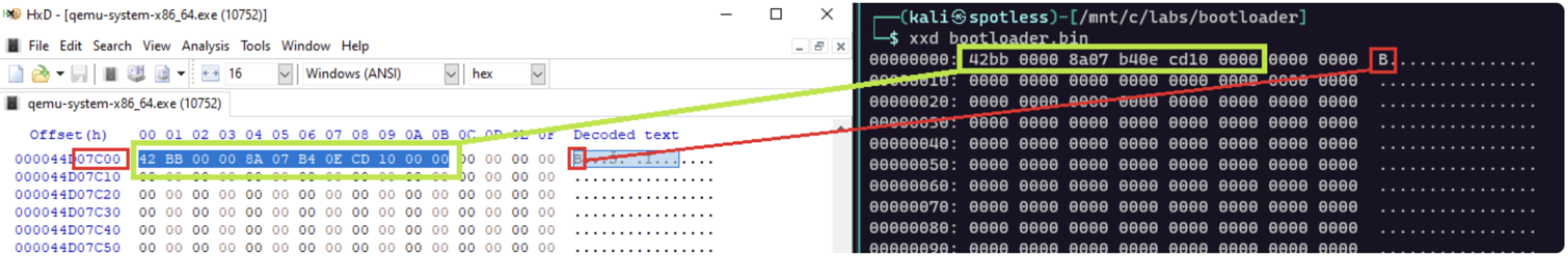


bootloader-x.asm

```
bits 16

; Define a label X that is a memory offset of the start of our code.
; It points to a character B.
x:
    db "B"

; Move offset of x to bx
mov bx, x

; Add 0x7c00 to bx - it's universally known that BIOS loads bootloaders to this locati
add bx, 0x7c00

; Move contents of bx to al
mov al, [bx]

; Prepare interrupt to print a character in TTY mode and issue the interrupt.
mov ah, 0x0e
int 0x10

times 510 - ($-$$) db 0
dw 0xaa55
```

https://www.ired.team/miscellaneous-reversing-forensics/windows-kernel-internals/writing-a-custom-bootloader

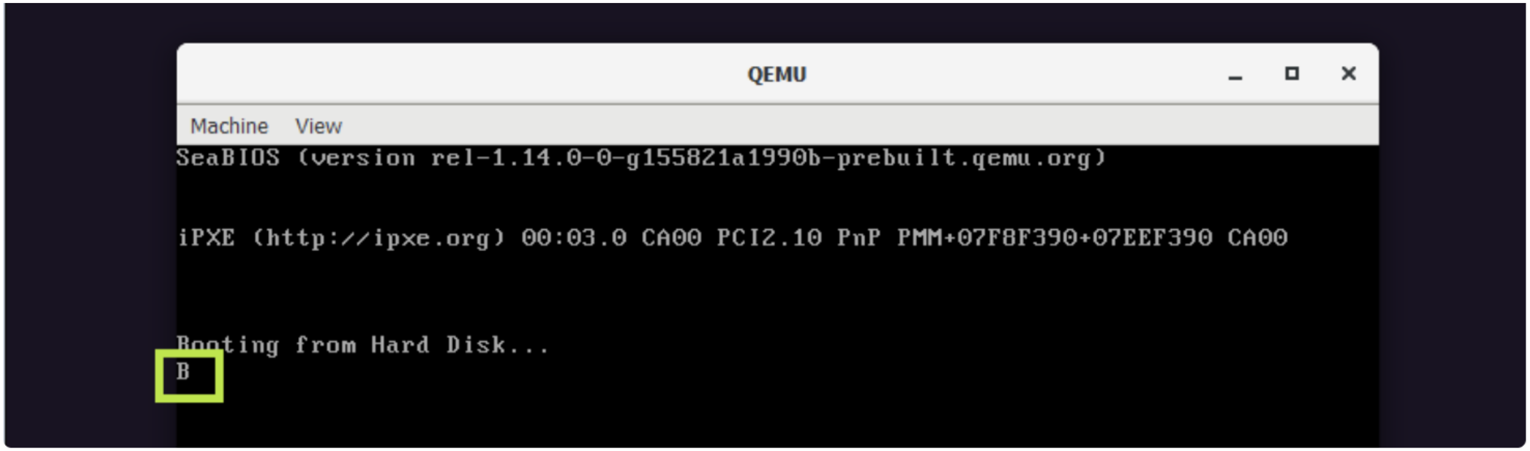
# MEMORIA LA BOOT



HxD - [qemu-system-x86_64.exe (10752)]
File Edit Search View Analysis Tools Window Help
16
Windows (ANSI)
hex
qemu-system-x86_64.exe (10752)

```
Offset(h)   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   Decoded text
000044D07C00 42 BB 00 00 8A 07 B4 0E CD 10 00 00 00 00 00 00   B»..Š.´.Í.......
000044D07C10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000044D07C20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000044D07C30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000044D07C40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000044D07C50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

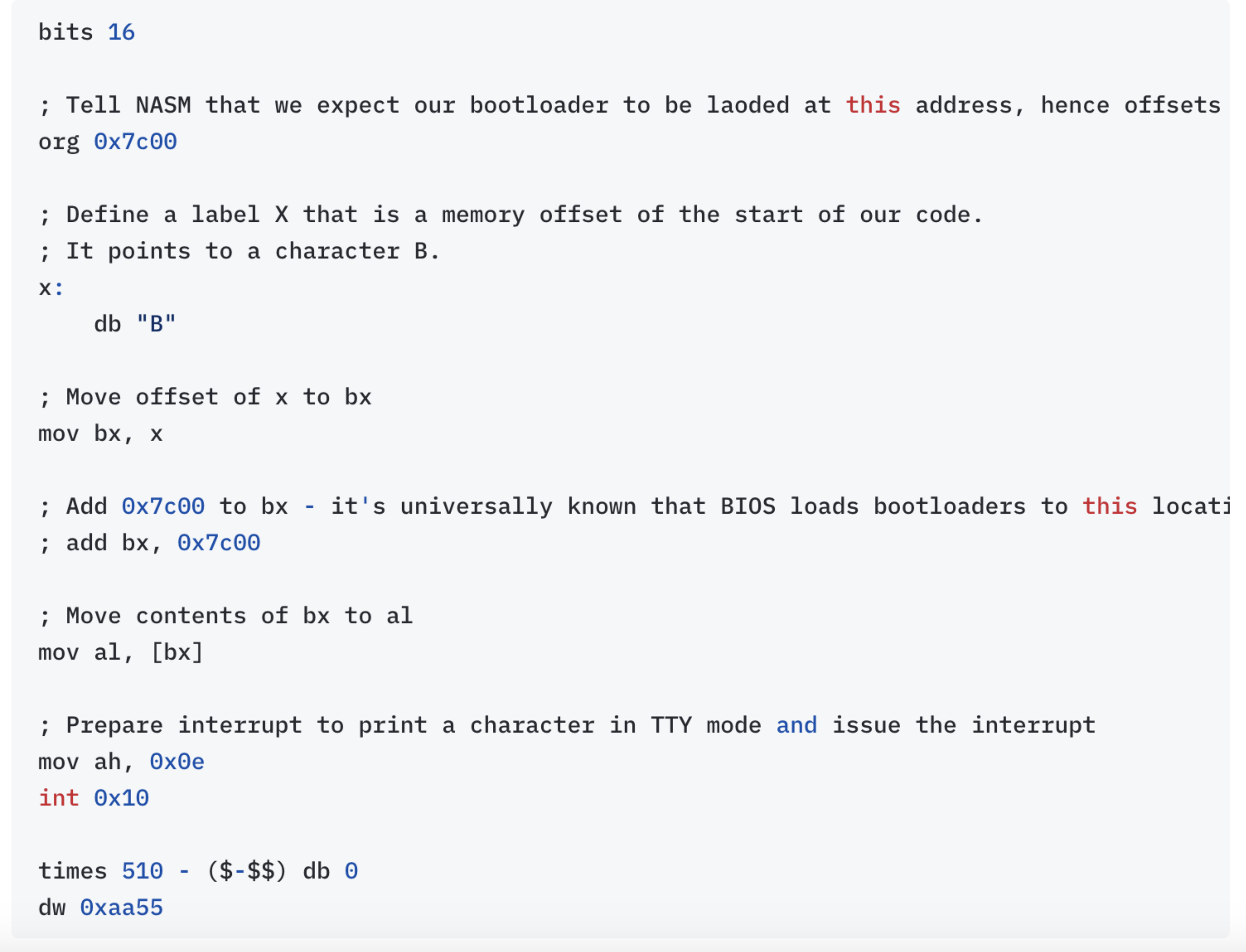
```
┌──(kali㉿spotless)-[/mnt/c/labs/bootloader]
└─$ xxd bootloader.bin
00000000: 42bb 0000 8a07 b40e cd10 0000 0000 0000  B...............
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```



```
SeaBIOS (version rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8F390+07EEF390 CA00


Booting from Hard Disk...
B
```

.

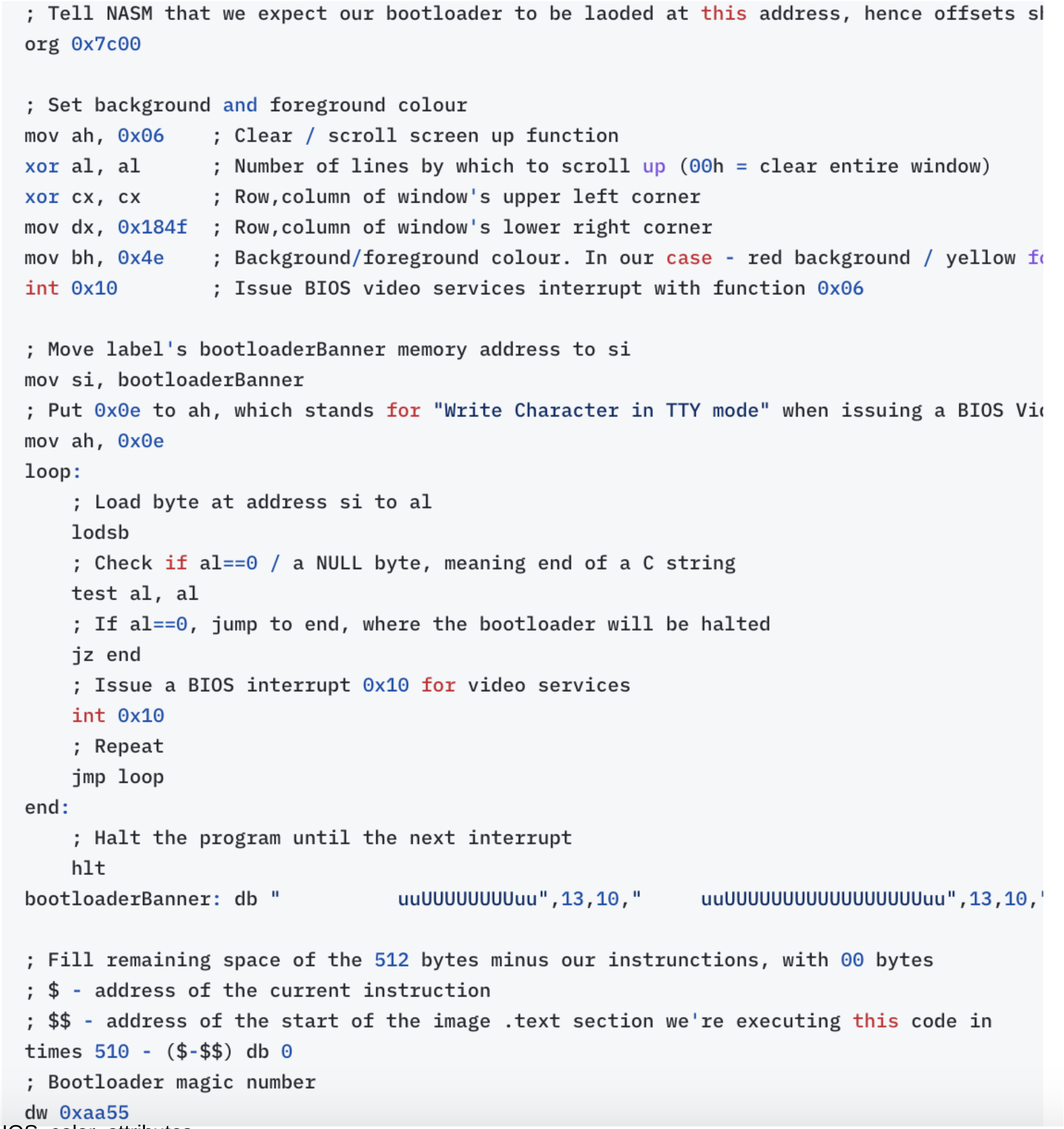
# AFIȘARE LA BOOT



```
bits 16

; Tell NASM that we expect our bootloader to be laoded at this address, hence offsets
org 0x7c00

; Define a label X that is a memory offset of the start of our code.
; It points to a character B.
x:
    db "B"

; Move offset of x to bx
mov bx, x

; Add 0x7c00 to bx - it's universally known that BIOS loads bootloaders to this locati
; add bx, 0x7c00

; Move contents of bx to al
mov al, [bx]

; Prepare interrupt to print a character in TTY mode and issue the interrupt
mov ah, 0x0e
int 0x10

times 510 - ($-$$) db 0
dw 0xaa55
```

https://en.wikipedia.org/wiki/BIOS_interrupt_call

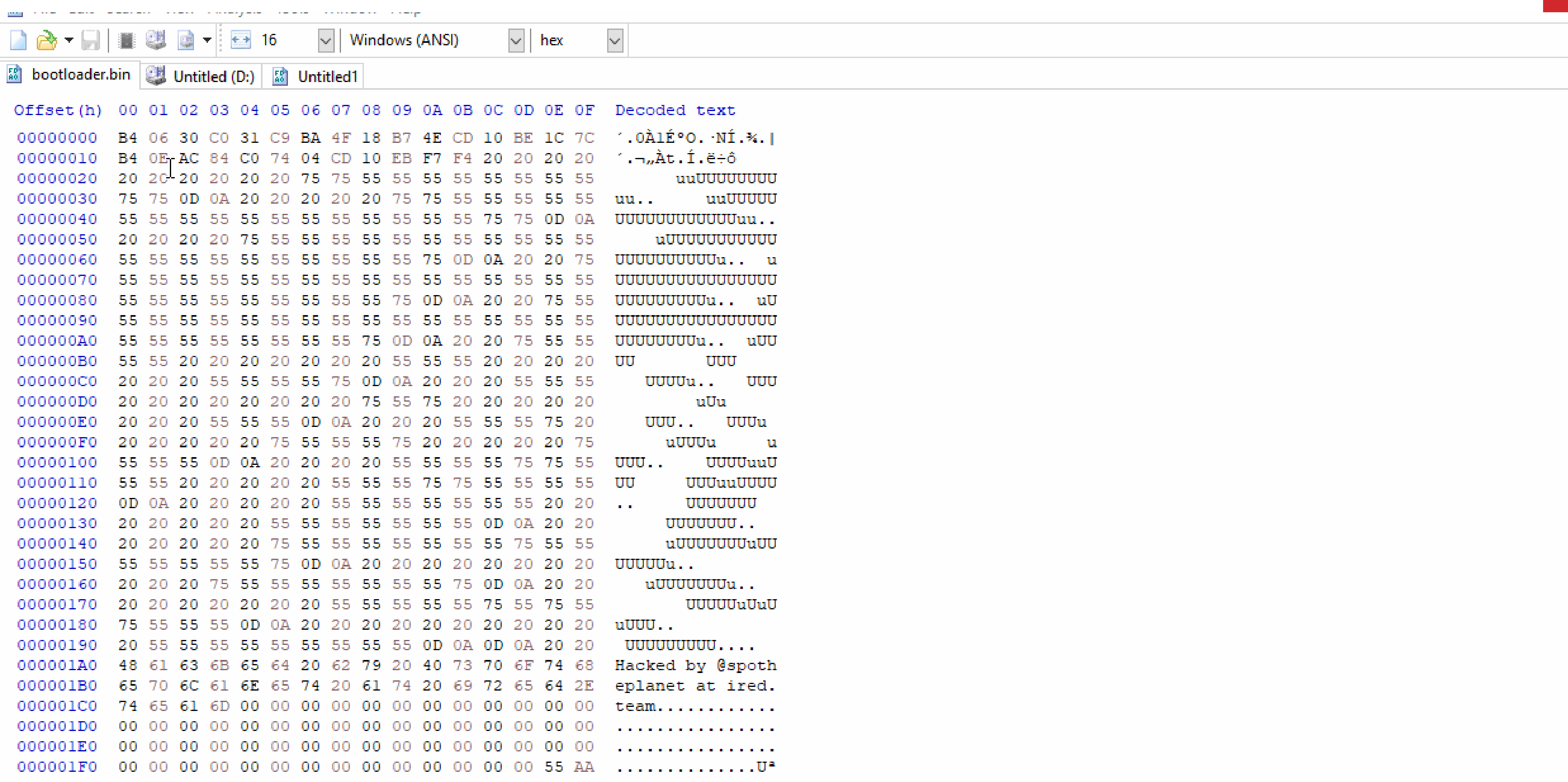
# AFIȘARE LA BOOT



```
; Tell NASM that we expect our bootloader to be laoded at this address, hence offsets sl
org 0x7c00

; Set background and foreground colour
mov ah, 0x06    ; Clear / scroll screen up function
xor al, al      ; Number of lines by which to scroll up (00h = clear entire window)
xor cx, cx      ; Row,column of window's upper left corner
mov dx, 0x184f  ; Row,column of window's lower right corner
mov bh, 0x4e    ; Background/foreground colour. In our case - red background / yellow fo
int 0x10        ; Issue BIOS video services interrupt with function 0x06

; Move label's bootloaderBanner memory address to si
mov si, bootloaderBanner
; Put 0x0e to ah, which stands for "Write Character in TTY mode" when issuing a BIOS Vic
mov ah, 0x0e
loop:
    ; Load byte at address si to al
    lodsb
    ; Check if al==0 / a NULL byte, meaning end of a C string
    test al, al
    ; If al==0, jump to end, where the bootloader will be halted
    jz end
    ; Issue a BIOS interrupt 0x10 for video services
    int 0x10
    ; Repeat
    jmp loop
end:
    ; Halt the program until the next interrupt
    hlt
bootloaderBanner: db "          uuUUUUUUUUuu",13,10,"     uuUUUUUUUUUUUUUUUUUuu",13,10,'

; Fill remaining space of the 512 bytes minus our instrunctions, with 00 bytes
; $ - address of the current instruction
; $$ - address of the start of the image .text section we're executing this code in
times 510 - ($-$$) db 0
; Bootloader magic number
dw 0xaa55
```

https://en.wikipedia.org/wiki/BIOS_color_attributes

# SECTORUL DE BOOT COPIAT



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  B4 06 30 C0 31 C9 BA 4F 18 B7 4E CD 10 BE 1C 7C  ´.0À1É°O.·NÍ.¾.|
00000010  B4 0E AC 84 C0 74 04 CD 10 EB F7 F4 20 20 20 20  ´.¬„Àt.Í.ë÷ô    
00000020  20 20 20 20 20 20 75 75 55 55 55 55 55 55 55 55        uuUUUUUUUU
00000030  75 75 0D 0A 20 20 20 20 20 75 75 55 55 55 55 55  uu..     uuUUUUU
00000040  55 55 55 55 55 55 55 55 55 55 55 55 75 75 0D 0A  UUUUUUUUUUUUuu..
00000050  20 20 20 20 75 55 55 55 55 55 55 55 55 55 55 55      uUUUUUUUUUU
00000060  55 55 55 55 55 55 55 55 55 55 75 0D 0A 20 20 75  UUUUUUUUUUu..  u
00000070  55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55  UUUUUUUUUUUUUUUU
00000080  55 55 55 55 55 55 55 55 55 75 0D 0A 20 20 75 55  UUUUUUUUUu..  uU
00000090  55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55  UUUUUUUUUUUUUUUU
000000A0  55 55 55 55 55 55 55 55 75 0D 0A 20 20 75 55 55  UUUUUUUUu..  uUU
000000B0  55 55 20 20 20 20 20 20 20 55 55 55 20 20 20 20  UU       UUU    
000000C0  20 20 20 55 55 55 55 75 0D 0A 20 20 20 55 55 55     UUUUu..   UUU
000000D0  20 20 20 20 20 20 20 20 75 55 75 20 20 20 20 20          uUu     
000000E0  20 20 20 55 55 55 0D 0A 20 20 20 55 55 55 75 20     UUU..   UUUu 
000000F0  20 20 20 20 20 75 55 55 55 75 20 20 20 20 20 75       uUUUu     u
00000100  55 55 55 0D 0A 20 20 20 20 55 55 55 55 75 75 55  UUU..    UUUUuuU
00000110  55 55 20 20 20 20 20 55 55 55 75 75 55 55 55 55  UU     UUUuuUUUU
00000120  0D 0A 20 20 20 20 20 55 55 55 55 55 55 55 20 20  ..     UUUUUUU  
00000130  20 20 20 20 20 55 55 55 55 55 55 55 0D 0A 20 20       UUUUUUU..  
00000140  20 20 20 20 20 75 55 55 55 55 55 55 55 75 55 55       uUUUUUUUuUU
00000150  55 55 55 55 55 75 0D 0A 20 20 20 20 20 20 20 20  UUUUUu..        
00000160  20 20 20 75 55 55 55 55 55 55 55 75 0D 0A 20 20     uUUUUUUUu..  
00000170  20 20 20 20 20 20 20 55 55 55 55 55 75 55 75 55         UUUUUuUuU
00000180  75 55 55 55 0D 0A 20 20 20 20 20 20 20 20 20 20  uUUU..          
00000190  20 55 55 55 55 55 55 55 55 55 0D 0A 0D 0A 20 20   UUUUUUUUU....  
000001A0  48 61 63 6B 65 64 20 62 79 20 40 73 70 6F 74 68  Hacked by @spoth
000001B0  65 70 6C 61 6E 65 74 20 61 74 20 69 72 65 64 2E  eplanet at ired.
000001C0  74 65 61 6D 00 00 00 00 00 00 00 00 00 00 00 00  team............
000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ..............Uª
```

# CE AM FĂCUT ASTĂZI

- am detaliat secvența de boot
- am folosit tool-ul qemu
- am scris un bootloader simplu

# DATA VIITOARE ...

- Evaluarea de la laborator

# LECTURĂ SUPLIMENTARĂ

- **Nick Blundell, https://www.cs.bham.ac.uk/~exr/lectures/opsys/10_11/lectures/os-dev.pdf**

- **Writing a Custom Bootloader, https://www.ired.team/miscellaneous-reversing-forensics/windows-kernel-internals/writing-a-custom-bootloader**

- **cfenollosa, os-tutorial**
  - https://github.com/cfenollosa/os-tutorial/tree/master/00-environment
  - https://github.com/cfenollosa/os-tutorial/tree/master/01-bootsector-barebones
  - https://github.com/cfenollosa/os-tutorial/tree/master/02-bootsector-print
  - https://github.com/cfenollosa/os-tutorial/tree/master/03-bootsector-memory

.

\0