

# Kurs administrowania systemem Linux

Zajęcia nr 10: Środowisko uruchomieniowe procesorów x86-64

Część 1: Legacy BIOS

Instytut Informatyki Uniwersytetu Wrocławskiego

8 maja 2025

# Czemu na Wielkanoc nie można uruchomić Linuksa?

# Czemu na Wielkanoc nie można uruchomić Linuksa?

**..a trzeciego dnia  
zastali pusty grub**



## CP/M

- Control Program/Monitor, potem Control Program for Microcomputers, 1974 – 1983
- Gary Arlen Kildall (19 maja 1942 – 11 lipca 1994)
- Digital Research, Inc. (Kalifornia 1974, przejęta przez Novell w 1991)
- PL/M (Programming Language for Microcomputers), 1973
- Procesory 8-bitowe: 8080/8085, Z80; potem 16/32-bitowe: 8086, 68k

## Struktura

- Basic Input/Output System (BIOS) — hardware abstraction layer
- Basic Disk Operating System (BDOS) — hardware independent
- Console Command Processor (CCP) — hardware independent

## Innowacja

- Model warstwowy i mały BIOS zależny od sprzętu dawały przenośność.

## DOS

- IBM PC, premiera 12 sierpnia 1981. Procesor Intel 8088.
- W 1980 IBM rozważał sportowanie CP/M na architekturę PC.
- Ostatecznie stworzenie systemu zlecono firmie Microsoft.
- DOS wzorowany na CP/M.
- Nowy system plików FAT.
- BIOS, wzorowany na CP/M, umieszczony w ROM na płycie głównej.

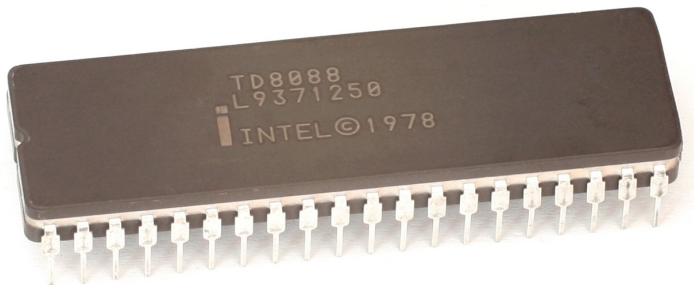
## Innowacja

- Hardware ma własny firmware dający Hardware Abstraction Layer.
- Option ROM: inne komponenty systemu (karta graficzna, sieciowa itp.) również dodają własny HAL do BIOS-u.

## Ewolucja

- Architektura x86 zawsze powiązana z HAL w firmwarze (BIOS, potem UEFI).
- Architektura ARM nie ma takiego powiązania.

# Procesor 8088



- Uproszczona wersja (8-bitowa magistrala adresowa) procesora 8086.
- Premiera: 1 lipca 1979 (8086: w połowie 1978).
- Architektura: 16 bitowa. Magistrala adresowa: 20-bitowa ( $2^{20} = 1\text{MiB}$ ).
- Obudowa: DIL 40-pin. Technologia: 3000 nm. Transistor count: 29 tys.
- Binarnie niezgodny z 8-bitowym procesorem 8080 (1974), ale *prawie* zgodny z nim *źródłowo*. 8080 zgodny źródłowo z 8008 (1972). Pierwszy mikroprocesor ever: 4004 (1971, 10000 nm, 2300 tranzystorów).

14 rejestrów 16-bitowych:

- Rejestry ogólnego przeznaczenia (GPR):  $aX = aH + aL$  dla  $a \in \{A, B, C, D\}$ .
- Rejestry indeksowe: **SI**, **DI**, **BP**, **SP** (*source, destination, base, stack*).
- Wskaźnik instrukcji: **IP**.
- Rejestry segmentowe: **CS**, **DS**, **SS**, **ES** (*code, data, stack, extra*).
- Rejestr statusu: **FLAGS**.

Adresowanie:  $(seg \ll 4) + base + index + offset$ , gdzie

- $seg \in \{CS, DS, SS, ES\}$ ,
- $base \in \{BX, BP\}$ ,
- $index \in \{SI, DI\}$ ,
- $offset = 16\text{-bit const.}$

# Rejstry procesora 8088

General purpose registers					AH								AL								AX
					BH								BL								BX
					CH								CL								CX
					DH								DL								DX
Index registers	0	0	0	0	SI								source index								
	0	0	0	0	DI								destination index								
	0	0	0	0	BP								base pointer								
	0	0	0	0	SP								stack pointer								
Program counter	0	0	0	0	IP								instruction pointer								
Segment registers					CS								code segment				0	0	0	0	
					DS								data segment				0	0	0	0	
					ES								extra segment				0	0	0	0	
					SS								stack segment				0	0	0	0	
Status register	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
					-	-	-	-	O	D	I	T	S	Z	-	A	-	P	-	C	FLAGS



## „640KiB wystarczy każdemu”

- Intel 8088 obsługuje adresy 20-bitowe ( $2^{20}\text{B} = 1\text{MiB}$ ).

## „640KiB wystarczy każdemu”

- Intel 8088 obsługuje adresy 20-bitowe ( $2^{20}\text{B} = 1\text{MiB}$ ).
- Przed uruchomieniem procesor ustawia rejestry CS (*code segment*) na ffff i IP (*instruction pointer*) na 0000, zatem pierwsza instrukcja wykonana przez procesor ma adres ffff0 ( $\text{address} = \text{CS} \ll 4 + \text{IP}$ ).

## „640KiB wystarczy każdemu”

- Intel 8088 obsługuje adresy 20-bitowe ( $2^{20}\text{B} = 1\text{MiB}$ ).
- Przed uruchomieniem procesor ustawia rejestry CS (*code segment*) na ffff i IP (*instruction pointer*) na 0000, zatem pierwsza instrukcja wykonana przez procesor ma adres ffff0 ( $\text{address} = \text{CS} \ll 4 + \text{IP}$ ).
- W PC/XT pamięć operacyjna (DRAM) zajmuje przestrzeń adresową 00000–9ffff (640KiB, tzw. *conventional memory*).

# „640KiB wystarczy każdemu”

- Intel 8088 obsługuje adresy 20-bitowe ( $2^{20}\text{B} = 1\text{MiB}$ ).
- Przed uruchomieniem procesor ustawia rejestry CS (*code segment*) na ffff i IP (*instruction pointer*) na 0000, zatem pierwsza instrukcja wykonana przez procesor ma adres ffff0 ( $\text{address} = \text{CS} \ll 4 + \text{IP}$ ).
- W PC/XT pamięć operacyjna (DRAM) zajmuje przestrzeń adresową 00000–9ffff (640KiB, tzw. *conventional memory*).
- Przestrzeń adresowa a0000–ffffff — UMB (*upper memory blocks*):
  - pamięć karty graficznej, np. Herkules: dwa monochromatyczne bufora  $720 \times 348$  w zakresie adresów: b0000–b75a8 i b8000–bf5a8,
  - pamięć ROM (program rozruchowy, tzw. BIOS): f0000–ffffff.

## „640KiB wystarczy każdemu”

- Intel 8088 obsługuje adresy 20-bitowe ( $2^{20}\text{B} = 1\text{MiB}$ ).
- Przed uruchomieniem procesor ustawia rejestry CS (*code segment*) na ffff i IP (*instruction pointer*) na 0000, zatem pierwsza instrukcja wykonana przez procesor ma adres ffff0 ( $\text{address} = \text{CS} \ll 4 + \text{IP}$ ).
- W PC/XT pamięć operacyjna (DRAM) zajmuje przestrzeń adresową 00000–9ffff (640KiB, tzw. *conventional memory*).
- Przestrzeń adresowa a0000–ffffff — UMB (*upper memory blocks*):
  - pamięć karty graficznej, np. Herkules: dwa monochromatyczne bufora  $720 \times 348$  w zakresie adresów: b0000–b75a8 i b8000–bf5a8,
  - pamięć ROM (program rozruchowy, tzw. BIOS): f0000–ffffff.
- Program rozruchowy BIOS:
  - procedury POST (*power-on self test*),
  - inicjalizacja urządzeń,
  - ładowanie programu z dyskietki lub dysku i przekazanie mu sterowania.

# BIOS i przerwania

Kod w ROM udostępnia uruchomionym programom procedury obsługi urządzeń: klawiatury, monitora, stacji dyskiety, dysków twardych itp. (których sam potrzebuje do załadowania programu) — stąd nazwa BIOS (*Basic Input/Output System*). To daje pierwszą warstwę abstrakcji między sprzętem i oprogramowaniem, np. zamiast samodzielnie obliczać

$$*(0xb0000 + ((y\&3)\ll 13) + (y\gg 2)*90 + (x\gg 3)) \mid= (1\ll (x\&7))$$

(zapal piksel o współrzędnych  $x$  i  $y$  — działa tylko dla karty Hercules), mamy:

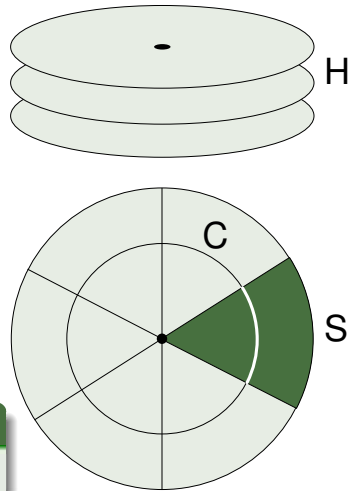
```
mov ah, 0xC ; funkcja 0xC w AH: write graphics pixel
mov al, 1   ; kolor piksela w AL
mov cx, y   ; współrzędna y w CX
mov dx, x   ; współrzędna x w DX
int 0x10    ; przerwanie 0x10: video services
```

# Numeracja sektorów dysku magnetycznego

- *Low-level formatting* — struktura sektorów zapisana przez *servowriter* przed zmontowaniem dysku.
- *Adres sektora*: C/H/S (*cylinder, głowica, sektor*).
- *Logical block addressing* — liniowa numeracja sektorów.
- *Ważne*: Ze względu na duży *seek time*
  - sąsiednie sektory *powinny* mieć kolejne adresy LBA (dlatego w *większości przypadków* sektor zajmuje *ustalone* miejsce na powierzchni plateru);
  - dane *powinny* być wyrównane do granicy cylindra.
- W dawnych czasach zwykle  $N_s = 63$ .

## Logical block addressing (LBA)

$$\text{LBA}(C, H, S) = (C \cdot N_h + H) \cdot N_s + (S - 1)$$



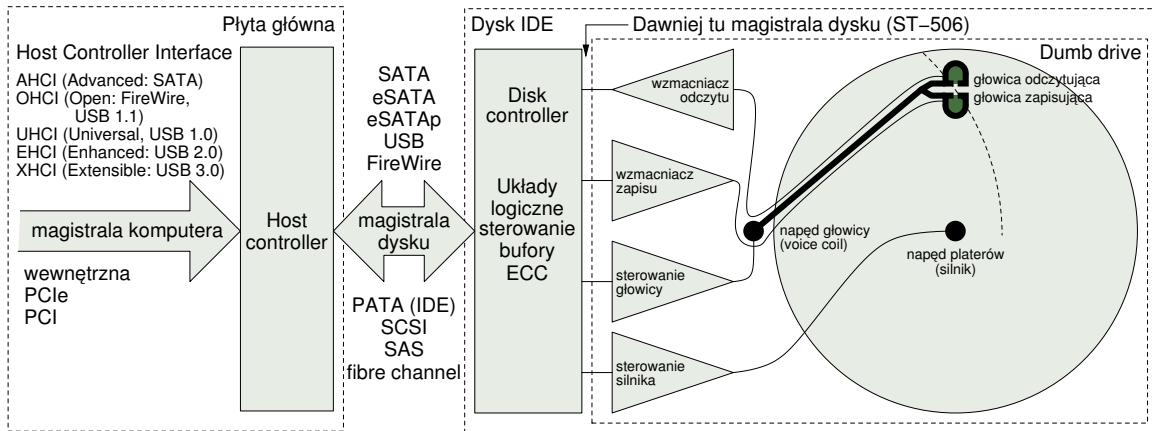
## Przerwanie int 13h: low-level disk services

```
mov ah, 0x2    ; funkcja nr 2: read sectors from drive
mov al, n      ; sectors to read count
mov ch, C      ; cylinder
mov cl, H      ; sector
mov dh, S      ; head
mov dl, 0x80    ; drive
mov es, ms     ; memory segment
mov bx, mo     ; memory offset
int 0x13        ; przerwanie 0x13: low-level disk services
```

drive	DL
A:	0x00
B:	0x01
C:	0x80
D:	0x81
CD ROM	0xe0



# Dyski IDE (Integrated Drive Electronics) — CHS jest fikcją



# Program rozruchowy BIOS-u: uruchomienie DOS-a

- Po zresetowaniu procesor wykonuje kod zapisany w ROM BIOS poczynając od adresu 0xffff0.
- Procedury POST (*power-on self test*).
- Inicjalizacja urządzeń.
- Dla dysków w kolejności *boot order* BIOS wykonuje poniższe czynności:
  - Ładuje sektor MBR (512B) do RAM pod adresem 0x07c00–0x07dff (wymagane co najmniej 32KiB RAM).
  - Jeśli odczyt przebiegł pomyślnie i dwa ostatnie bajty załadowanego sektora to 0x55aa, wtedy ustawia IP na 7c00, CS na 0000 i przekazuje sterowanie do właśnie załadowanego programu. W przeciwnym przypadku przechodzi do następnego dysku.
- Załadowany program nie musi być przesuwalny (zawsze się uruchamia w ustalonym miejscu), ale może mieć co najwyżej 446B.
- Uruchomiony program może (musi!) korzystać z usług BIOS-u za pomocą przerwań.

20-bitowa przestrzeń adresowa

**1MiB = 1024KiB = 1048576B**

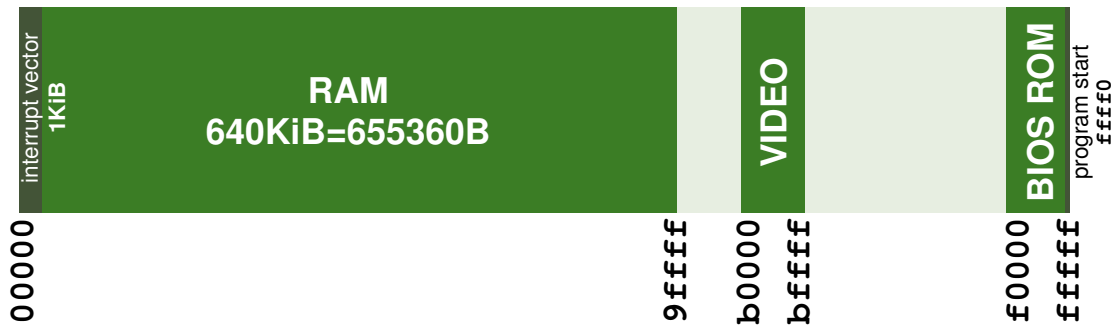
0000

ffff

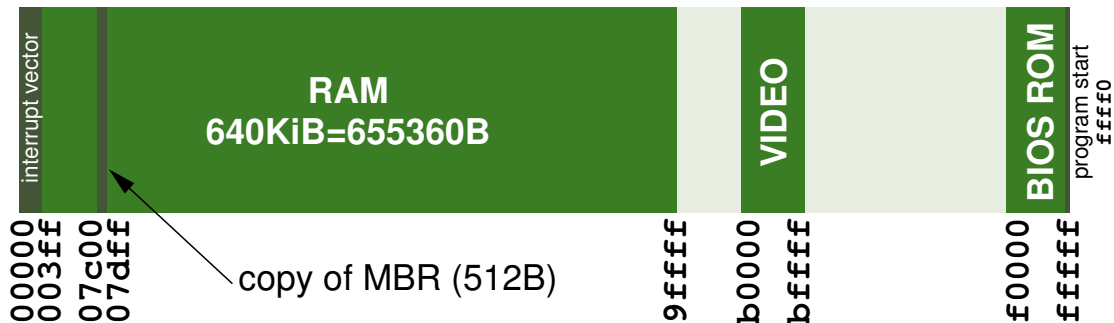
## Wyróżnione adresy procesora 8088



## Mapa pamięci PC/XT



## BIOS ładuje program z dysku



**1978** 8086 i 8088

**1982** 80286: *on-chip* MMU, *protected mode* — wsparcie dla systemów wielozadaniowych (Unix, Xenix, Minix, OS/2)

**1985** 80386 (IA-32): 32-bit, PMMU (*paging*)

**1989** 80486: *pipelining*

**1993** Pentium (80586): *superscalar*, 64-bit data bus, SMP

**1995** Pentium Pro: *out-of-order execution*

**2001** Itanium (IA-64): 64 bit, EPIC

**2003** Athlon 64 (x86-64): 64-bit od AMD

**2005** Pentium 4: 64-bit od Intel

# Procesory w komputerach PC, które używałem/używam

nazwa	premiera	rok	pin	arch	nm	tranzystorów	zegar MHz	pamięć
8088	1979	1989–1991	40	16	3000	29 000	4.77	320 KiB
80486SX	1992	2001–2002	168	32	1000	1 185 000	20	4 MiB
AMD K6	1997	2002–2008	321	32	350	8 800 000	166	64 MiB
Celeron 1400	2002	2006–teraz	370	32	130	44 000 000	1400	256 MiB
Core Duo E8400	2008	2008–teraz	775	64	45	410 000 000	$3000 \times 2$	2 GiB
Core i3-3217U	2012	2013–teraz	1023	64	22	970 000 000	$1800 \times 2/4$	4 GiB
Core i5-3230M	2013	2021–teraz	988	64	22	1 200 000 000	$2600 \times 2/4$	4 GiB
Core i5-6600	2015	2016–teraz	1168	64	14	1 350 000 000	$3300 \times 4$	16 GiB
Celeron N3050	2015	2017–teraz	1170	64	14	?	$1600 \times 2$	2 GiB



# „Ontogeneza jest parafrazą filogenezy”

- Współczesne procesory x86-64 zaczynają pracę zasadniczo tak samo jak 8088.
- Procesor startuje w trybie rzeczywistym (*real address mode*), w którym symuluje oryginalne adresowanie 20-bitowe.
- Oryginalny adres `ffff0` procesor przelicza na adres 32-bitowy `fffffff0` (4GiB minus 16B).
- Kontroler pamięci (MMU, most północny) mapuje z powrotem 32-bitowe adresy `ffff0000–fffffff0` na adresy `f0000–fffff` pamięci flash BIOS (tzw. *BIOS shadowing*).
- Dopiero uruchomiony program wykonuje rozkaz włączający tryb 32- lub 64-bitowy i wirtualizację pamięci.

# Master Boot Record

hex	dec	rozmiar	Przeznaczenie
0- 2	0- 2	3	Bootstrap code area part 1
3- 59	3- 89	87	BIOS parameter block
5a-1b7	90-439	350	Bootstrap code area part 2
1b8-1bb	440-443	4	Disk signature
1bc-1bd	444-445	2	0000 (rw) lub 5a5a (ro)
1be-1cd	446-461	16	Partition entry no. 1
1ce-1dd	462-477	16	Partition entry no. 2
1de-1ed	478-493	16	Partition entry no. 3
1ee-1fd	494-509	16	Partition entry no. 4
1fe-1ff	510-511	2	55aa (bootable) lub 0000

# Partition entry

hex	rozmiar	Znaczenie
0-0	1	0x80 (bootable), 0x00 (inactive)
1-3	3	CHS pierwszego sektora
4-4	1	typ partycji
5-7	3	CHS ostatniego sektora
8-b	4	LBA pierwszego sektora
c-f	4	liczba sektorów

Kodowanie CHS w bajtach 1-3 i 5-7

H7	H6	H5	H4	H3	H2	H1	H0	C9	C8	S5	S4	S3	S2	S1	S0	C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

$$0 \leq H \leq 255, 0 \leq C \leq 1023, 0 \leq S \leq 63.$$

# Typy partycji DOS/MBR

00 Empty	24 NEC DOS	81 Minix/old Linux	bf Solaris
01 FAT12	27 Hidden NTFS Windows RE	82 Linux swap/Solaris	c1 DRDOS/sec FAT12
02 XENIX root	39 Plan 9	83 Linux	c4 DRDOS/sec FAT16
03 XENIX usr	3c PartitionMagic	84 OS/2 hidden C:	c6 DRDOS/sec FAT32
04 FAT16 < 32M	40 Venix 80286	85 Linux extended	c7 Syrinx
05 Extended	41 PPC PReP Boot	86 NTFS volume set	da Non-FS data
06 FAT16	42 SFS	87 NTFS volume set	db CP/M / CTOS
07 HPFS/NTFS/exFAT	4d QNX4.x	88 Linux plaintext	de Dell Utility
08 AIX	4e QNX4.x 2nd part	8e Linux LVM	df BootIt
09 AIX bootable	4f QNX4.x 3rd part	93 Amoeba	e1 DOS access
0a OS/2 Boot Manager	50 OnTrack DM	94 Amoeba BBT	e3 DOS R/O
0b W95 FAT32	51 OnTrack DM6 Aux	9f BSD/OS	e4 SpeedStor
0c W95 FAT32 (LBA)	52 CP/M	a0 IBM Thinkpad hibernate	eb BeOS fs
0e W95 FAT16 (LBA)	53 OnTrack DM6 Aux	a5 FreeBSD	ee GPT
0f W95 Ext'd (LBA)	54 OnTrackDM6	a6 OpenBSD	ef EFI (FAT-12/16/32)
10 OPUS	55 EZ-Drive	a7 NeXTSTEP	f0 Linux/PA-RISC boot
11 Hidden FAT12	56 Golden Bow	a8 Darwin UFS	f1 SpeedStor
12 Compaq diagnostics	5c Priam Edisk	a9 NetBSD	f4 SpeedStor
14 Hidden FAT16 < 32M	61 SpeedStor	ab Darwin boot	f2 DOS secondary
16 Hidden FAT16	63 GNU HURD	af HFS / HFS+	fb VMware VMFS
17 Hidden HPFS/NTFS	64 Novell Netware	b7 BSDI fs	fc VMware VMKCORE
18 AST SmartSleep	65 Novell Netware	b8 BSDI swap	fd Linux raid auto
1b Hidden W95 FAT32	70 DiskSecure Multiboot	bb Boot Wizard hidden	fe LANstep
1c Hidden W95 FAT32	75 PC/IX	be Solaris boot	ff BBT
1e Hidden W95 FAT16	80 Old Minix		

- Partycja zawierająca wewnątrz *partycje logiczne*.
- Typ: 0f (dawniej 05).
- Partycje logiczne pozwalają na podział dysku na więcej niż cztery partycje.
- Wewnątrz partycji rozszerzonej lista wiązana partycji logicznych.
- Nagłówek EBR partycji logicznej: 4 czterobajtowe pola plus padding.
- Wiele programów nie rozumie partycji rozszerzonych. . .

# Przykładowy dysk z zainstalowanym Linuksem

```
fdisk -l /dev/sda
```

```
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 4096 bytes
```

```
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

```
Disklabel type: dos
```

```
Disk identifier: 0xa8793cb8
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	133119	131072	64M	83	Linux
/dev/sda2		133120	16777215	16644096	8G	83	Linux

$$2048 = 2^{11}, 131072 = 2^{17}, 16777216 = 2^{24}$$

# Grub2 w Master Boot Record

## Master Boot Record

hexdump -C -n 512 /dev/sda

```
00000000 eb 63 90 00 00 00 00 00 00 00 00 00 00 00 00 |.c.....|
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000050 00 00 00 00 00 00 00 00 00 00 80 01 00 00 |.....|
00000060 00 00 00 00 ff fa 90 90 f6 c2 80 74 05 f6 c2 70 |.....t...p|
00000070 74 02 b2 80 ea 79 7c 00 00 31 c0 8e d8 8e d0 bc |t....y|..1.....|
00000080 00 20 fb a0 64 7c 3c ff 74 02 88 c2 52 bb 17 04 |...d|<.t...R...|
00000090 f6 07 03 74 06 be 88 7d e8 17 01 be 05 7c b4 41 |...t...}.....|.A|
000000a0 bb aa 55 cd 13 5a 52 72 3d 81 fb 55 aa 75 37 83 |..U..ZRR=..U.u7.|
000000b0 e1 01 74 32 31 c0 89 44 04 40 88 44 ff 89 44 02 |..t21..D.@.D..D.|
000000c0 c7 04 10 00 66 8b 1e 5c 7c 66 89 5c 08 66 8b 1e |...f...\\f\\.f..|
000000d0 60 7c 66 89 5c 0c c7 44 06 00 70 b4 42 cd 13 72 |'|f\\.D..p.B..r|
000000e0 05 bb 00 70 eb 76 b4 08 cd 13 73 0d 5a 84 d2 0f |...p.v....s.Z...|
000000f0 83 d0 00 be 93 7d e9 82 00 66 0f b6 c6 88 64 ff |....}...f....d.|
00000100 40 66 89 44 04 0f b6 d1 c1 e2 02 88 e8 88 f4 40 |@f.D.....@|
00000110 89 44 08 0f b6 c2 c0 e8 02 66 89 04 66 a1 60 7c |.D.....f..f..'|
00000120 66 09 c0 75 4e 66 a1 5c 7c 66 31 d2 66 f7 34 88 |f..uNf.\\f1.f.4.|
00000130 d1 31 d2 66 f7 74 04 3b 44 08 7d 37 fe c1 88 c5 |.1.f.t.;D.}7....|
00000140 30 c0 c1 e8 02 08 c1 88 d0 5a 88 c6 bb 00 70 8e |0.....Z...p.|
00000150 c3 31 db b8 01 02 cd 13 72 1e 8c c3 60 1e b9 00 |.1.....r.....|
00000160 01 8e db 31 f6 bf 00 80 8e c6 fc f3 a5 1f 61 ff |...1.....a..|
00000170 26 5a 7c be 8e 7d eb 03 be 9d 7d e8 34 00 be a2 |&Z|...}....}.4...|
00000180 7d e8 2e 00 cd 18 eb fe 47 52 55 42 20 00 47 65 |}.....GRUB..Ge|
00000190 6f 6d 00 48 61 72 64 20 44 69 73 6b 00 52 65 61 |om.Hard Disk.Rea|
000001a0 64 00 20 45 72 72 6f 72 0d 0a 00 bb 01 00 b4 0e |d. Error.....|
000001b0 cd 10 ac 3c 00 75 f4 c3 b8 3c 79 a8 00 00 80 20 |...<.u...<y....|
000001c0 21 00 83 49 01 08 00 08 00 00 00 00 02 00 00 49 |!..I.....I|
000001d0 02 08 83 55 01 14 00 08 02 00 00 f8 fd 00 00 00 |...U.....|
000001e0 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 55 aa |.....U..|
```

# Grub2 w Master Boot Record

## BIOS parameter block

[illegible]



# Grub2 w Master Boot Record

## GRUB2 part 1: 353B

00000000	eb 63 90	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.c.....
00000010	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	
00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	
00000050	00 00 00 00 00 00 00 00 00 00 00 80 01 00 00 00	.....	
00000060	00 00 00 00 ff fa 90 90 f6 c2 80 74 05 f6 c2 70	.....t...p	
00000070	74 02 b2 80 ea 79 7c 00 00 31 c0 8e d8 8e d0 bc	t...y ..1....	
00000080	00 20 fb a0 64 7c 3c ff 74 02 88 c2 52 bb 17 04	...d <.t...R...	
00000090	f6 07 03 74 06 be 88 7d e8 17 01 be 05 7c b4 41	...t...}..... .A	
000000a0	bb aa 55 cd 13 5a 52 72 3d 81 fb 55 aa 75 37 83	..U..ZrR=..U.u7.	
000000b0	e1 01 74 32 31 c0 89 44 04 40 88 44 ff 89 44 02	..t21..D.@.D..D.	
000000c0	c7 04 10 00 66 8b 1e 5c 7c 66 89 5c 08 66 8b 1e	...f... f.\.f...	
000000d0	60 7c 66 89 5c 0c c7 44 06 00 70 b4 42 cd 13 72	' f.\.D..p.B..r	
000000e0	05 bb 00 70 eb 76 b4 08 cd 13 73 0d 5a 84 d2 0f	...p.v....s.Z...	
000000f0	83 d0 00 be 93 7d e9 82 00 66 0f b6 c6 88 64 ff	.....}...f....d.	
00000100	40 66 89 44 04 0f b6 d1 c1 e2 02 88 e8 88 f4 40	@f.D.....@	
00000110	89 44 08 0f b6 c2 c0 e8 02 66 89 04 66 a1 60 7c	.D.....f..f.'	
00000120	66 09 c0 75 4e 66 a1 5c 7c 66 31 d2 66 f7 34 88	f..uNf.\ f1.f.4.	
00000130	d1 31 d2 66 f7 74 04 3b 44 08 7d 37 fe c1 88 c5	.1.f.t.;D.}7....	
00000140	30 c0 c1 e8 02 08 c1 88 d0 5a 88 c6 bb 00 70 8e	0.....Z....p.	
00000150	c3 31 db b8 01 02 cd 13 72 1e 8c c3 60 1e b9 00	.1.....r.....	
00000160	01 8e db 31 f6 bf 00 80 8e c6 fc f3 a5 1f 61 ff	...1.....a.	
00000170	26 5a 7c be 8e 7d eb 03 be 9d 7d e8 34 00 be a2	&Z ..}....}.4...	
00000180	7d e8 2e 00 cd 18 eb fe 47 52 55 42 20 00 47 65	}.....GRUB .Ge	
00000190	6f 6d 00 48 61 72 64 20 44 69 73 6b 00 52 65 61	om.Hard Disk.Rea	
000001a0	64 00 20 45 72 72 6f 72 0d 0a 00 bb 01 00 b4 0e	d. Error.....	
000001b0	cd 10 ac 3c 00 75 f4 c3 b8 3c 79 a8 00 00 80 20	...<.u...<y....	
000001c0	21 00 83 49 01 08 00 08 00 00 00 00 02 00 00 49	!...I.....I	
000001d0	02 08 83 55 01 14 00 08 02 00 00 f8 fd 00 00 00	...U.....	
000001e0	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 55 aa	.....U.	

# Grub2 w Master Boot Record

## Disk UUID 4B (UUID=a8793cb8)

```
00000000 eb 63 90 00 00 00 00 00 00 00 00 00 00 00 00 |.c.....|
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000050 00 00 00 00 00 00 00 00 00 00 80 01 00 00 |.....|
00000060 00 00 00 00 ff fa 90 90 f6 c2 80 74 05 f6 c2 70 |.....t...p|
00000070 74 02 b2 80 ea 79 7c 00 00 31 c0 8e d8 8e d0 bc |t...y|..1.....|
00000080 00 20 fb a0 64 7c 3c ff 74 02 88 c2 52 bb 17 04 |...d|<.t...R...|
00000090 f6 07 03 74 06 be 88 7d e8 17 01 be 05 7c b4 41 |...t...}.....|.A|
000000a0 bb aa 55 cd 13 5a 52 72 3d 81 fb 55 aa 75 37 83 |..U..ZRR=..U.u7.|
000000b0 e1 01 74 32 31 c0 89 44 04 40 88 44 ff 89 44 02 |..t21..D.@.D..D.|
000000c0 c7 04 10 00 66 8b 1e 5c 7c 66 89 5c 08 66 8b 1e |...f...\\f...f...|
000000d0 60 7c 66 89 5c 0c c7 44 06 00 70 b4 42 cd 13 72 |'|f...D...p.B...r|
000000e0 05 bb 00 70 eb 76 b4 08 cd 13 73 0d 5a 84 d2 0f |...p.v....s.Z...|
000000f0 83 d0 00 be 93 7d e9 82 00 66 0f b6 c6 88 64 ff |.....}...f....d.|
00000100 40 66 89 44 04 0f b6 d1 c1 e2 02 88 e8 88 f4 40 |@f.D.....}....@|
00000110 89 44 08 0f b6 c2 c0 e8 02 66 89 04 66 a1 60 7c |.D.....f...f...'|
00000120 66 09 c0 75 4e 66 a1 5c 7c 66 31 d2 66 f7 34 88 |f..uNf...\\f1.f.4.|
00000130 d1 31 d2 66 f7 74 04 3b 44 08 7d 37 fe c1 88 c5 |.1.f.t.;D.}7....|
00000140 30 c0 c1 e8 02 08 c1 88 d0 5a 88 c6 bb 00 70 8e |0.....Z...p...|
00000150 c3 31 db b8 01 02 cd 13 72 1e 8c c3 60 1e b9 00 |.1.....r...'|
00000160 01 8e db 31 f6 bf 00 80 8e c6 fc f3 a5 1f 61 ff |...1.....a...|
00000170 26 5a 7c be 8e 7d eb 03 be 9d 7d e8 34 00 be a2 |&Z|...}....}.4...|
00000180 7d e8 2e 00 cd 18 eb fe 47 52 55 42 20 00 47 65 |}.....GRUB...Ge|
00000190 6f 6d 00 48 61 72 64 20 44 69 73 6b 00 52 65 61 |om.Hard Disk.Rea|
000001a0 64 00 20 45 72 72 6f 72 0d 0a 00 bb 01 00 b4 0e |d. Error.....|
000001b0 cd 10 ac 3c 00 75 f4 c3 b8 3c 79 a8 00 00 80 20 |...<.u...<y....|
000001c0 21 00 83 49 01 08 00 08 00 00 00 00 02 00 00 49 |!..I.....I...|
000001d0 02 08 83 55 01 14 00 08 02 00 00 f8 fd 00 00 00 |...U.....|
000001e0 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 55 aa |.....U...|
```

# Grub2 w Master Boot Record

0000 (or 5a5a if read-only)

```

00000000 eb 63 90 00 00 00 00 00 00 00 00 00 00 00 00 |.c.....
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000050 00 00 00 00 00 00 00 00 00 00 80 01 00 00 |.....
00000060 00 00 00 00 ff fa 90 90 f6 c2 80 74 05 f6 c2 70 |.....t...p
00000070 74 02 b2 80 ea 79 7c 00 00 31 c0 8e d8 8e d0 bc |t....y|.1...
00000080 00 20 fb a0 64 7c 3c ff 74 02 88 c2 52 bb 17 04 |...d|<.t...R...
00000090 f6 07 03 74 06 be 88 7d e8 17 01 be 05 7c b4 41 |...t...}|....|.A
000000a0 bb aa 55 cd 13 5a 52 72 3d 81 fb 55 aa 75 37 83 |..U..ZRr=.U.u7.
000000b0 e1 01 74 32 31 c0 89 44 04 40 88 44 ff 89 44 02 |.t21..D.@.D.D.
000000c0 c7 04 10 00 66 8b 1e 5c 7c 66 89 5c 08 66 8b 1e |...f...\\f.f..
000000d0 60 7c 66 89 5c 0c b7 44 06 00 70 b4 42 cd 13 72 |'|f.f...D..p.B..r
000000e0 05 bb 00 70 eb 76 b4 08 cd 13 73 0d 5a 84 d2 0f |...p.v....s.Z...
000000f0 83 d0 00 be 93 7d e9 82 00 66 0f b6 c6 88 64 ff |...}|...f....d
00000100 40 66 89 44 04 0f b6 d1 c1 e2 02 88 e8 88 f4 40 |@f.D.....@
00000110 89 44 08 0f b6 c2 c0 e8 02 66 89 04 66 a1 60 7c |.D.....f.f.f.'|
00000120 66 09 c0 75 e6 a1 5c 7c 66 31 d2 66 f7 34 88 |f..uNf.f1.f.4.
00000130 d1 31 d2 66 f7 74 04 3b 44 08 7d 37 fe c1 88 c5 |.1.f.t.;D?7...
00000140 30 c0 c1 e8 02 08 c1 88 d0 5a 88 c6 bb 00 70 8e |0.....Z....p.
00000150 c3 31 db b8 01 02 cd 13 72 1e 8c c3 60 1e b9 00 |.1.....r...'
00000160 01 8e db 31 f6 bf 00 80 8e c6 fc f3 a5 1f 61 ff |...1.....a.
00000170 26 5a 7c be 8e 7d eb 03 be 9d 7d e8 34 00 be a2 |&Z|..}|....}.4...
00000180 7d e8 2e 00 cd 18 eb fe 47 52 55 42 20 00 47 65 |}|.....GRUB .Ge
00000190 6f 6d 00 48 61 72 64 20 44 69 73 6b 00 52 65 61 |om.Hard Disk.Rea
000001a0 64 00 20 45 72 72 6f 72 0d 0a 00 bb 01 00 b4 0e |d. Error.....
000001b0 cd 10 ac 3c 00 75 f4 c3 b8 3c 79 ab 00 00 80 20 |...<.u...<y..
000001c0 21 00 83 49 01 08 00 08 00 00 00 00 02 00 00 49 |!..I.....I
000001d0 02 08 83 55 01 14 00 08 02 00 00 f8 fd 00 00 00 |...U.....
000001e0 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 55 aa |.....U

```

# Grub2 w Master Boot Record

Partition 1: flag 08 (bootable), type 83 (Linux)

start LBA 00000800h = 2048, size 00020000h = 131072 blocks

00000000	eb 63 90 00 00 00 00 00	00 00 00 00 00 00 00 00	.c.....
00000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000050	00 00 00 00 00 00 00 00	00 00 00 80 01 00 00 00	.....
00000060	00 00 00 00 ff fa 90 90	f6 c2 80 74 05 f6 c2 70	.....t...p
00000070	74 02 b2 80 ea 79 7c 00	00 31 c0 8e d8 8e d0 bc	t...y ..1.....
00000080	00 20 fb a0 64 7c 3c ff	74 02 88 c2 52 bb 17 04	...d <.t...R...
00000090	f6 07 03 74 06 be 88 7d	e8 17 01 be 05 7c b4 41	...t...}..... .A
000000a0	bb aa 55 cd 13 5a 52 72	3d 81 fb 55 aa 75 37 83	..U..ZRR=..U.u7..
000000b0	e1 01 74 32 31 c0 89 44	04 40 88 44 ff 89 44 02	..t21..D.@.D..D..
000000c0	c7 04 10 00 66 8b 1e 5c	7c 66 89 5c 08 66 8b 1e	...f..\\f..f..
000000d0	60 7c 66 89 5c 0c c7 44	06 00 70 b4 42 cd 13 72	' f..\\D..p.B..r
000000e0	05 bb 00 70 eb 76 b4 08	cd 13 73 0d 5a 84 d2 0f	...p.v....s.Z...
000000f0	83 d0 00 be 93 7d e9 82	00 66 0f b6 c6 88 64 ff	.....}...f.....d..
00000100	40 66 89 44 04 0f b6 d1	c1 e2 02 88 e8 88 f4 40	@f.D.....}...@
00000110	89 44 08 0f b6 c2 c0 e8	02 66 89 04 66 a1 60 7c	.D.....f..f..'
00000120	66 09 c0 75 4e 66 a1 5c	7c 66 31 d2 66 f7 34 88	f..uNf..\\f1.f.4..
00000130	d1 31 d2 66 f7 74 04 3b	44 08 7d 37 fe c1 88 c5	.1.f.t.;D.}7....
00000140	30 c0 c1 e8 02 08 c1 88	d0 5a 88 c6 bb 00 70 8e	0.....Z...p..
00000150	c3 31 db b8 01 02 cd 13	72 1e 8c c3 60 1e b9 00	.1.....r.....
00000160	01 8e db 31 f6 bf 00 80	8e c6 fc f3 a5 1f 61 ff	...1.....a..
00000170	26 5a 7c be 8e 7d eb 03	be 9d 7d e8 34 00 be a2	&Z ..}....}.4...
00000180	7d e8 2e 00 cd 18 eb fe	47 52 55 42 20 00 47 65	}.....GRUB..Ge
00000190	6f 6d 00 48 61 72 64 20	44 69 73 6b 00 52 65 61	om.Hard Disk.Rea
000001a0	64 00 20 45 72 72 6f 72	0d 0a 00 bb 01 00 b4 0e	d. Error.....
000001b0	cd 10 ac 3c 00 75 f4 c3	b8 3c 79 a8 00 00 80 20	...<.u...<y....
000001c0	21 00 83 49 01 08 00 08	00 00 00 02 00 00 49	!..I.....I..
000001d0	02 08 83 55 01 14 00 08	02 00 00 f8 fd 00 00 00	...U.....
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	.....U..

# Grub2 w Master Boot Record

## Partition 2: type 83 (Linux)

start LBA 00020800h = 133120, size 00fdf800h = 16644096 blocks

00000000	eb 63 90 00 00 00 00 00	00 00 00 00 00 00 00 00	.c.....
00000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000050	00 00 00 00 00 00 00 00	00 00 00 80 01 00 00 00	.....
00000060	00 00 00 00 ff fa 90 90	f6 c2 80 74 05 f6 c2 70	.....t...p
00000070	74 02 b2 80 ea 79 7c 00	00 31 c0 8e d8 8e d0 bc	t...y ..1.....
00000080	00 20 fb a0 64 7c 3c ff	74 02 88 c2 52 bb 17 04	...d <.t...R...
00000090	f6 07 03 74 06 be 88 7d	e8 17 01 be 05 7c b4 41	...t...}..... .A
000000a0	bb aa 55 cd 13 5a 52 72	3d 81 fb 55 aa 75 37 83	..U..ZRR=..U.u7..
000000b0	e1 01 74 32 31 c0 89 44	04 40 88 44 ff 89 44 02	..t21..D.@.D..D..
000000c0	c7 04 10 00 66 8b 1e 5c	7c 66 89 5c 08 66 8b 1e	...f..\\f..f..
000000d0	60 7c 66 89 5c 0c c7 44	06 00 70 b4 42 cd 13 72	' f..\\D..p.B..r
000000e0	05 bb 00 70 eb 76 b4 08	cd 13 73 0d 5a 84 d2 0f	...p.v....s.Z...
000000f0	83 d0 00 be 93 7d e9 82	00 66 0f b6 c6 88 64 ff	.....}...f.....d..
00000100	40 66 89 44 04 0f b6 d1	c1 e2 02 88 e8 88 f4 40	@f.D.....}...@
00000110	89 44 08 0f b6 c2 c0 e8	02 66 89 04 66 a1 60 7c	.D.....f..f..'
00000120	66 09 c0 75 4e 66 a1 5c	7c 66 31 d2 66 f7 34 88	f..uNf..\\f1.f.4..
00000130	d1 31 d2 66 f7 74 04 3b	44 08 7d 37 fe c1 88 c5	.1.f.t.;D.}7....
00000140	30 c0 c1 e8 02 08 c1 88	d0 5a 88 c6 bb 00 70 8e	0.....Z...p..
00000150	c3 31 db b8 01 02 cd 13	72 1e 8c c3 60 1e b9 00	.1.....r.....
00000160	01 8e db 31 f6 bf 00 80	8e c6 fc f3 a5 1f 61 ff	...1.....a..
00000170	26 5a 7c be 8e 7d eb 03	be 9d 7d e8 34 00 be a2	&Z ..}....}.4...
00000180	7d e8 2e 00 cd 18 eb fe	47 52 55 42 20 00 47 65	}.....GRUB..Ge
00000190	6f 6d 00 48 61 72 64 20	44 69 73 6b 00 52 65 61	om.Hard Disk.Rea
000001a0	64 00 20 45 72 72 6f 72	0d 0a 00 bb 01 00 b4 0e	d. Error.....
000001b0	cd 10 ac 3c 00 75 f4 c3	b8 3c 79 a8 00 00 80 20	...<.u...<y....
000001c0	21 00 83 49 01 08 00 08	00 00 00 02 00 00 00 00	!..I.....}...I
000001d0	02 08 83 55 01 14 00 08	02 00 00 f8 fd 00 00 00	...U.....
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	.....U..

# Grub2 w Master Boot Record

## Partition 3 (absent)

00000000	eb 63 90 00 00 00 00 00	00 00 00 00 00 00 00 00	.c.....
00000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000050	00 00 00 00 00 00 00 00	00 00 00 80 01 00 00 00	.....
00000060	00 00 00 00 ff fa 90 90	f6 c2 80 74 05 f6 c2 70	.....t...p
00000070	74 02 b2 80 ea 79 7c 00	00 31 c0 8e d8 8e d0 bc	t...y ..1.....
00000080	00 20 fb a0 64 7c 3c ff	74 02 88 c2 52 bb 17 04	...d <.t....R...
00000090	f6 07 03 74 06 be 88 7d	e8 17 01 be 05 7c b4 41	...t...}..... .A
000000a0	bb aa 55 cd 13 5a 52 72	3d 81 fb 55 aa 75 37 83	..U..ZRR=..U.u7..
000000b0	e1 01 74 32 31 c0 89 44	04 40 88 44 ff 89 44 02	..t21..D.@.D..D..
000000c0	c7 04 10 00 66 8b 1e 5c	7c 66 89 5c 08 66 8b 1e	...f...\\f...f...
000000d0	60 7c 66 89 5c 0c c7 44	06 00 70 b4 42 cd 13 72	' f...D...p.B...r
000000e0	05 bb 00 70 eb 76 b4 08	cd 13 73 0d 5a 84 d2 0f	...p.v....s.Z...
000000f0	83 d0 00 be 93 7d e9 82	00 66 0f b6 c6 88 64 ff	.....}...f.....d..
00000100	40 66 89 44 04 0f b6 d1	c1 e2 02 88 e8 88 f4 40	@f.D.....}...@
00000110	89 44 08 0f b6 c2 c0 e8	02 66 89 04 66 a1 60 7c	.D.....f...f...'
00000120	66 09 c0 75 4e 66 a1 5c	7c 66 31 d2 66 f7 34 88	f..uNf...\\f1.f.4..
00000130	d1 31 d2 66 f7 74 04 3b	44 08 7d 37 fe c1 88 c5	.1.f.t.;D.}7....
00000140	30 c0 c1 e8 02 08 c1 88	d0 5a 88 c6 bb 00 70 8e	0.....Z...p...
00000150	c3 31 db b8 01 02 cd 13	72 1e 8c c3 60 1e b9 00	.1.....r...'
00000160	01 8e db 31 f6 bf 00 80	8e c6 fc f3 a5 1f 61 ff	...1.....a...
00000170	26 5a 7c be 8e 7d eb 03	be 9d 7d e8 34 00 be a2	&Z ...}....}.4...
00000180	7d e8 2e 00 cd 18 eb fe	47 52 55 42 20 00 47 65	}.....GRUB...Ge
00000190	6f 6d 00 48 61 72 64 20	44 69 73 6b 00 52 65 61	om.Hard Disk.Rea
000001a0	64 00 20 45 72 72 6f 72	0d 0a 00 bb 01 00 b4 0e	d. Error.....
000001b0	cd 10 ac 3c 00 75 f4 c3	b8 3c 79 a8 00 00 80 20	...<.u...<y....
000001c0	21 00 83 49 01 08 00 08	00 00 00 00 02 00 00 49	!.I.....I...
000001d0	02 08 83 55 01 14 00 08	02 00 00 f8 fd 00 00 00	...U.....
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	.....U...

# Grub2 w Master Boot Record

## Partition 4 (absent)

00000000	eb 63 90 00 00 00 00 00	00 00 00 00 00 00 00 00	.c.....
00000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000050	00 00 00 00 00 00 00 00	00 00 00 80 01 00 00 00	.....
00000060	00 00 00 00 ff fa 90 90	f6 c2 80 74 05 f6 c2 70	.....t...p
00000070	74 02 b2 80 ea 79 7c 00	00 31 c0 8e d8 8e d0 bc	t...y ..1.....
00000080	00 20 fb a0 64 7c 3c ff	74 02 88 c2 52 bb 17 04	...d <.t....R...
00000090	f6 07 03 74 06 be 88 7d	e8 17 01 be 05 7c b4 41	...t...}..... .A
000000a0	bb aa 55 cd 13 5a 52 72	3d 81 fb 55 aa 75 37 83	..U..ZRR=..U.u7..
000000b0	e1 01 74 32 31 c0 89 44	04 40 88 44 ff 89 44 02	..t21..D.@.D..D..
000000c0	c7 04 10 00 66 8b 1e 5c	7c 66 89 5c 08 66 8b 1e	...f... f.\.f...
000000d0	60 7c 66 89 5c 0c c7 44	06 00 70 b4 42 cd 13 72	' f.\.D..p.B..r
000000e0	05 bb 00 70 eb 76 b4 08	cd 13 73 0d 5a 84 d2 0f	...p.v....s.Z...
000000f0	83 d0 00 be 93 7d e9 82	00 66 0f b6 c6 88 64 ff	.....}...f.....d..
00000100	40 66 89 44 04 0f b6 d1	c1 e2 02 88 e8 88 f4 40	@f.D.....}...@
00000110	89 44 08 0f b6 c2 c0 e8	02 66 89 04 66 a1 60 7c	.D.....f..f.'
00000120	66 09 c0 75 4e 66 a1 5c	7c 66 31 d2 66 f7 34 88	f..uNf.\ f1.f.4..
00000130	d1 31 d2 66 f7 74 04 3b	44 08 7d 37 fe c1 88 c5	.1.f.t.;D.}7....
00000140	30 c0 c1 e8 02 08 c1 88	d0 5a 88 c6 bb 00 70 8e	0.....Z...p...
00000150	c3 31 db b8 01 02 cd 13	72 1e 8c c3 60 1e b9 00	.1.....r.....
00000160	01 8e db 31 f6 bf 00 80	8e c6 fc f3 a5 1f 61 ff	...1.....a...
00000170	26 5a 7c be 8e 7d eb 03	be 9d 7d e8 34 00 be a2	&Z ..}....}.4...
00000180	7d e8 2e 00 cd 18 eb fe	47 52 55 42 20 00 47 65	}.....GRUB..Ge
00000190	6f 6d 00 48 61 72 64 20	44 69 73 6b 00 52 65 61	om.Hard Disk.Rea
000001a0	64 00 20 45 72 72 6f 72	0d 0a 00 bb 01 00 b4 0e	d. Error.....
000001b0	cd 10 ac 3c 00 75 f4 c3	b8 3c 79 a8 00 00 80 20	...<.u...<y....
000001c0	21 00 83 49 01 08 00 08	00 00 00 00 02 00 00 49	!..I.....I...
000001d0	02 08 83 55 01 14 00 08	02 00 00 f8 fd 00 00 00	...U.....
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	.....U...

# Grub2 w Master Boot Record

## Magic number 55aa (bootable disk)

00000000	eb 63 90 00 00 00 00 00	00 00 00 00 00 00 00 00	.c.....
00000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00000050	00 00 00 00 00 00 00 00	00 00 00 80 01 00 00 00	.....
00000060	00 00 00 00 ff fa 90 90	f6 c2 80 74 05 f6 c2 70	.....t...p
00000070	74 02 b2 80 ea 79 7c 00	00 31 c0 8e d8 8e d0 bc	t....y ..1.....
00000080	00 20 fb a0 64 7c 3c ff	74 02 88 c2 52 bb 17 04	...d <.t...R...
00000090	f6 07 03 74 06 be 88 7d	e8 17 01 be 05 7c b4 41	...t...}..... .A
000000a0	bb aa 55 cd 13 5a 52 72	3d 81 fb 55 aa 75 37 83	..U..ZRR=..U.u7.
000000b0	e1 01 74 32 31 c0 89 44	04 40 88 44 ff 89 44 02	..t21..D.@.D..D.
000000c0	c7 04 10 00 66 8b 1e 5c	7c 66 89 5c 08 66 8b 1e	...f...\\f...f...
000000d0	60 7c 66 89 5c 0c c7 44	06 00 70 b4 42 cd 13 72	' f...D...p.B...r
000000e0	05 bb 00 70 eb 76 b4 08	cd 13 73 0d 5a 84 d2 0f	...p.v....s.Z...
000000f0	83 d0 00 be 93 7d e9 82	00 66 0f b6 c6 88 64 ff	.....}...f.....d.
00000100	40 66 89 44 04 0f b6 d1	c1 e2 02 88 e8 88 f4 40	@f.D.....}...@
00000110	89 44 08 0f b6 c2 c0 e8	02 66 89 04 66 a1 60 7c	.D.....f...f...'
00000120	66 09 c0 75 4e 66 a1 5c	7c 66 31 d2 66 f7 34 88	f..uNf...\\f1.f.4.
00000130	d1 31 d2 66 f7 74 04 3b	44 08 7d 37 fe c1 88 c5	.1.f.t.;D.}7....
00000140	30 c0 c1 e8 02 08 c1 88	d0 5a 88 c6 bb 00 70 8e	0.....Z...p...
00000150	c3 31 db b8 01 02 cd 13	72 1e 8c c3 60 1e b9 00	.1.....r...'
00000160	01 8e db 31 f6 bf 00 80	8e c6 fc f3 a5 1f 61 ff	...1.....a...
00000170	26 5a 7c be 8e 7d eb 03	be 9d 7d e8 34 00 be a2	&Z ...}....}.4...
00000180	7d e8 2e 00 cd 18 eb fe	47 52 55 42 20 00 47 65	}.....GRUB...Ge
00000190	6f 6d 00 48 61 72 64 20	44 69 73 6b 00 52 65 61	om.Hard Disk.Rea
000001a0	64 00 20 45 72 72 6f 72	0d 0a 00 bb 01 00 b4 0e	d. Error.....
000001b0	cd 10 ac 3c 00 75 f4 c3	b8 3c 79 a8 00 00 80 20	...<.u...<y....
000001c0	21 00 83 49 01 08 00 08	00 00 00 00 02 00 00 49	!..I.....I...
000001d0	02 08 83 55 01 14 00 08	02 00 00 f8 fd 00 00 00	...U.....
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	.....U...



# Deasemblowanie MBR-a

```
objdump -D -m i386 -b binary -M addr16,intel /dev/sda
```

```
                _start:
00: eb 63                jmp 0x65
02: 90                    nop
                   . = _start + 0x5a
kernel address:
5a: 00 80                .word 0x8000
5c: 01 00 00 00          .long 1
60: 00 00 00 00          .long 0
64: ff                    .byte 0xff
65: fa                    cli
66: 90                    nop
67: 90                    nop
68: f6 c2 80             test    dl,0x80
6b: 74 05                je      0x72
6d: f6 c2 70             test    dl,0x70
70: 74 02                je      0x74
72: b2 80                mov     dl,0x80
```

- BIOS pozwala załadować  $\leq 466$  bajtów kodu z MBR. Dalej oprogramowanie musi radzić sobie samo.
- 466 bajtów wystarczało do załadowania i uruchomienia DOS-u.
- W przypadku współczesnych systemów operacyjnych lepiej wprowadzić warstwę pośrednią:

BIOS loader  $\rightarrow$  boot loader bootstrap 1  $\rightarrow \dots \rightarrow$  boot loader bootstrap  $i \rightarrow$  OS kernel

## Rodzaje programów pośredniczących

- *Boot loader*
- *Boot manager*

# Typowy pośredni program ładujący na przykładzie GRUB-a

## Uruchamianie GRUB-a

- *Faza 1*: W MBR 16-bitowy kod korzystający z `int 13h` w celu załadowania reszty *jądra*.
- *Faza 1a*: Reszta jądra GRUB-a: dostęp do systemu plików zawierającego pliki potrzebne do dalszej pracy.
- *Faza 2*: Działanie pełnego programu po załadowaniu *modułów*.
- Jeśli faza 2 się nie uda, to mamy grub `rescue`.
- W przypadku MBR ograniczenia rozmiaru jądra (31KiB) wymusiły budowę modułarną.

## Miejsce na jądro GRUB-a

- *Wyrównanie początku partycji* powoduje powstanie pustego obszaru między MBR i pierwszą partycją. Tam jest zapisywane jądro.
- Klasycznie wyrównanie do początku cylindra. Zwykle cylinder ma 63 sektory. Stąd 62 sektory (31KiB) wolne.
- Obecnie wyrównanie do 1MiB, stąd 2047 sektorów wolnych. Jądro GRUB-a ma zwykle 50KiB.

## Jądro GRUB-a

- Składane przez program `grub-mkimage`.
- Należy dodać co najmniej moduły potrzebne do zamontowania systemu plików z `rootfs-em` GRUB-a (sterowniki itp.).
- Można dodać wczesny plik konfiguracyjny, inne dodatkowe pliki (np. czcionki).
- Jądro i `rootfs` gruba instaluje program `grub-install`, ale można użyć też zwykłych programów `dd` i `rsync`.
- Jeśli wszystko, co potrzeba, dodamy do jądra, to `rootfs-a` można nie instalować.

## Rootfs GRUB-a zwykle zawiera:

- główny plik konfiguracji `grub.cfg`,
- plik `grubenv` (1024 B, jedyny plik modyfikowany przez GRUB-a),
- katalog z *modułami* `*.mod` dla danej architektury (np. `i386-pc/`, `x86_64-efi/`),
- katalog `fonts/` z plikami `*.pf2`,
- katalog `locale/` z plikami `*.mo`.

Można stworzyć konfigurację *multiarch* (np. `i386-pc + x86_64-efi`).

## Plik konfiguracyjny GRUB-a

- Opisuje menu i sposób uruchamiania systemów operacyjnych.
- Język przypomina `bash`-a.
- Bardzo dobra dokumentacja (projekt GNU!).
- Warto pisać samodzielnie.
- **ZŁO:** program `grub-mkconfig(8)` i infrastruktura `update-grub`.
- **ZŁO:** `os-prober`.

## „GRUB to osobny system operacyjny”

- Wygodna konfiguracja komputera — osobne partycje: bootloader, OS-y, dane (w kilku częściach); wszystko konfigurowane niezależnie.
- GRUB-a lepiej instalować ręcznie niż używać instalatora któregoś z OS-ów.
- OS-y też lepiej instalować ręcznie niż korzystać z instalatorów.
- Warto mieć więcej niż jedną partycję z OS-em.

# Tryby pracy procesora x86-64

**Long mode** 64-bitowy tryb chroniony. Możliwość wykonywania programów trybie kompatybilności w:

- 32-bitowym trybie chronionym,
- 16-bitowym trybie chronionym.

**Legacy protected mode** 32-bitowy tryb chroniony. Procesor działa tak jak IA-32 (i386). Możliwość wykonywania programów w trybie kompatybilności w:

- 16-bitowym trybie chronionym,
- 16-bitowym trybie wirtualnym (emulacja trybu rzeczywistego).

**Legacy real mode** 16-bitowy tryb rzeczywisty. Procesor działa tak jak 8086.

**System Management Mode** specjalny tryb wykonania kodu z BIOS-u.

Procesor rozpoczyna pracę w 16-bitowym trybie rzeczywistym.

# Włączenie trybu chronionego

Aby przejść do trybu chronionego należy:

- wyłączyć przerwania (w tym niemaskowalne),
- włączyć linię A20 (wyłączenie sprzętowego ograniczenia pamięci do 1MiB),
- załadować globalną tablicę deskryptorów (segmentowanie pamięci),
- ustawić bit PE (*protection enable*) w rejestrze CR0,
- wykonać daleki skok do segmentu kodu wykonywanego w trybie chronionym.

Powrót do trybu rzeczywistego polega na odwróceniu powyższych czynności.

# Włączenie trybu 64-bitowego

Aby włączyć *long mode* należy:

- wyłączyć stronicowanie,
- ustawić bit PAE (*physical address extension*) w rejestrze CR4,
- załadować PML4 (*page map level 4*) do rejestru CR3,
- ustawić flagę EFER.LME w MSR (*model specific register*) 0xC0000080,
- włączyć stronicowanie.

Teraz procesor jest w *long mode* w trybie 32-bitowym. Przełączanie długości adresów kodu: w globalnej tablicy deskryptorów w segmencie kodu ustawić bity L i D/B:

tryb	L	D/B
16-bit	0	0
32-bit	0	1
64-bit	1	0



# Zjeść ciasteczko i mieć ciasteczko: x32

Zalety i wady *long mode*:

- + 16 rejestrów 64-bitowych (zamiast 8 rejestrów 32-bitowych),
- + 16 rejestrów 128-bitowych XMM (dedykowanych dla SSE) zamiast ośmiu,
- + większy zbiór instrukcji SSE (Streaming SIMD Extension),
- adresy zajmują 8 bajtów (zamiast 4) — większy *footprint* w pamięci i *cache*.

Ale można wykonywać programy w *long mode* w trybie 32-bitowym!

Nowe ABI Linuksa (2012):

- wymaga jądra 3.4 i glibc 2.16,
- trzeba włączyć za pomocą parametru jądra `syscall.x32=y`.

Przebiegiem:

- zużycie pamięci mniejsze o około 30% (niż w trybie 64-bitowym),
- wydajność większa o około 40% (niż w trybie 32-bitowym).

Zobacz: <https://wiki.debian.org/X32Port>.

# Rozruch komputera z BIOS-em

- Ładuje kod *bootloadera* z MBR i przekazuje mu sterowanie w 16-bitowym trybie rzeczywistym.
- Udostępnia usługi zarządzania dyskami i urządzeniami wejścia/wyjścia poprzez przerwania (`int13h`, `int10h` itp).
- *Bootloader* (opcjonalnie) sam przełącza procesor do *protected mode*. Po przełączeniu usługi BIOS-u stają się niedostępne.
- W grub-pc (BIOS) istnieje możliwość uruchomienia jądra zarówno w trybie rzeczywistym, jak i 32-bitowym chronionym:
  - `linux` — uruchamia jądro w trybie chronionym 32-bitowym, stronicowanie jest wyłączone;
  - `linux16` — uruchamia jądro w trybie rzeczywistym 16-bitowym.

## Było w poprzednim odcinku: *Bootloader* uruchamia jądro

*Bootloader* umieszcza w pamięci:

- plik z jądrem systemu operacyjnego *vmlinuz* (skompresowany),
- (opcjonalnie) plik z *initramfs* (skompresowany),
- strukturę *boot\_params* (tzw. *zero page*).

i przekazuje jądro sterowanie.

- Plik z jądrem ma format *zImage* (*zlib-compressed image*) lub *bzImage* (*big zImage*,  $\geq 512\text{KiB}$ ). W nagłówku zawiera kod wykonywalny, którego zadaniem jest przejęcie maszyny i rozpakowanie kodu jądra.
- Może być uruchomiony w trybie rzeczywistym, chronionym lub długim.

## Było w poprzednim odcinku: Jądro zaczyna pracę

- `start()` z modułu `head.S` (przejmuje sterowanie, ew. włącza tryb chroniony lub długi, stronicowanie wyłączone),
- `decompress_kernel()` z modułu `misc.c` (wypakowuje „prawdziwe” jądro `vmlinux`),
- `startup_32()` z modułu `head.S` (inicjalizuje tablice stron i włącza stronicowanie),
- `start_kernel()` z modułu `head.S` (włącza *scheduler*, `cpu_idle()` i wątki jądra, uruchamia pierwszy proces (*init*) z przestrzeni użytkownika).

# Przekazywanie sterowania pomiędzy *bootloaderem* i jądrem

## The Linux/x86 Boot Protocol

- Opisuje sposób przekazywania sterowania od *bootloadera* do jądra.
- <https://www.kernel.org/doc/Documentation/x86/boot.txt>
- Obecna wersja: 2.12 (jądro 3.8).
- Ważna struktura: `boot_params` (zob. `zero-page.txt`). Zawiera m. in. mapę pamięci, tzw. E820 (funkcja `e820h` przerwania 15h).

## Multiboot Specification (FSF 1995)

- Dowolny zgodny *bootloader* może uruchomić dowolny zgodny system operacyjny.
- GRUB jest wzorcową implementacją.
- Obecna wersja 0.6.96 (2009).
- Zgodne *bootloadery*: GRUB 2, Xen, OpenBIOS, freldr (*bootloader* ReactOSa), *bootloader* MirBSD.
- Zgodne systemy: GNU Hurd, MINIX 3, NetBSD, Illumos (fork OpenSolarisa), AROS Research Operating System, CapROS, SkyOS, Syllable. Linux **nie** jest zgodny.

# BIOS z lat 1980-tych w komputerze z XXI wieku

- Tablica partycji nieadekwatna do dzisiejszych dysków.
  - Kodowanie CHS (24 bity) pozwala zaadresować  $\leq 8\text{GiB}$  — od lat nie używane.
  - Numeracja LBA (32 bity) pozwala zaadresować  $\leq 2\text{TiB}$ .
  - Podział tylko na 4 partycje (plus niewygodne partycje rozszerzone).
- Ładowanie programu z MBR i konieczność jego *bootstrapowania* z sektorów zapisanych poza partycjami.
- Konieczność uruchamiania programu w trybie 16-bitowym.