ΕΚΚΙΝΗΣΗ Η/Υ ΜΕ WINDOWS XP(NT) ΕΓΚΑΤΕΣΤΗΜΕΝΑ

Όλα ξεκινούν όταν πιέζουμε το γνωστό κουμπί στο κουτί του υπολογιστής μας έτσι ώστε να τροφοδοτηθεί με ρεύμα και να αρχίσει η όλη διαδικασία εκκίνησης.
Σε πρώτη φάση το BIOS της μητρικής αναλαμβάνει να εκτελέσει κάποιους ελέγχους έτσι ώστε να επιβεβαιώσει την παρουσία και την ορθή λειτουργία των απαραίτητων συσκευών του συστήματος. Στη συνέχεια θα ψάξει για συσκευές από τις οποίες μπορεί να εκκινήσει(Συνήθως σκληροί δίσκοι αλλά όχι μόνο. Εάν ρίξουμε μια ματιά στο BIOS της μητρικής μπορούμε να δούμε συσκευές από τις οποίες μπορεί να εκκινήσει το σύστημα) το λειτουργικό και θα μεταφέρει τον έλεγχο εκεί.
Οι Η/Υ που βασίζονται στην x86 αρχιτεκτονική ξεκινούν τη διαδικασία εκκίνησης από δίσκους που περιέχουν το MBR.

MBR

Είναι ο πρώτος βασικός τομέας για την εκκίνηση. Η κύρια εγγραφή εκκίνησης (MBR) βρίσκεται στην αρχή κάθε διαμορφωμένου φυσικού δίσκου (στον πρώτο φυσικό τομέα του δίσκου) έξω από κάθε partition και είναι μεγέθους 512 bytes .Δημιουργείται όταν ο δίσκος διαμερίζεται.
Η δομή του MBR είναι η παρακάτω:

A **master boot record** (**MBR**), or **partition sector**, is the 512-byte boot sector that is the first sector ("LBA Sector 0") of a partitioned data storage device such as a hard disk. (The boot sector of a non-partitioned device is a Volume Boot Record. These are usually different, although it is possible to create a record that acts as both; it is called a multi boot record.) The MBR may be used for one or more of the following:

- Holding a disk's primary partition table.[2]
- Bootstrapping operating systems, after the computer's BIOS passes execution to machine code instructions contained within the MBR.
- Uniquely identifying individual disk media, with a 32-bit *disk signature*; even though it may never be used by the machine the disk is running on.[3][4][5][6]

Due to the broad popularity of IBM PC-compatible computers, this type of MBR is widely used, to the extent of being supported by and incorporated into other computer types including newer cross-platform standards for bootstrapping and partitioning.[*citation needed*]

Structure of a Master Boot Record

Address			Description		Size in bytes
Hex	Oct	Dec			
0000	0000	0	Code Area		**440** (max. 446)
01B8	0670	440	Optional Disk signature		4
01BC	0674	444	Usually Nulls; 0x0000		2
01BE	0676	446	**Table of primary partitions** (Four 16-byte entries, IBM Partition Table scheme)		**64**
01FE	0776	510	55h	MBR signature; 0xAA55[1]	**2**
01FF	0777	511	AAh		
MBR, total size: 446 + 64 + 2 =					**512**

Όπως βλέπουμε στην παραπάνω εικόνα τα πρώτα 440 bytes περιέχουν τον κώδικα εκκίνησης που θα εκτελεστεί .
Τα επόμενα 4 bytes είναι το optional disk signature και τα άλλα 2 bytes είναι συνήθως 0.
Ακολουθεί το Table of primary partition μεγέθους 64 bytes με τέσσερις εγγραφές 16 byte η κάθε μία όπου εγγράφονται οι πληροφορίες για τα πρωτεύοντα partitions του δίσκου. Τέλος, στα 2 τελευταία bytes είναι η υπογραφή του MBR που είναι 0xAA55.
Το BIOS διαβάζει τον τομέα αυτό και αν εντοπίσει την υπογραφή φορτώνει τον τομέα στη μνήμη και δίνει τον έλεγχο στον κώδικα του sector.
Σε γενικές γραμμές ο κώδικας του MBR εκτελεί τα παρακάτω:

- Ανιχνεύει το ενεργό (active) διαμέρισμα στο Table of primary partition το οποίο πρέπει να είναι και μοναδικό.
- Μεταφέρει τον έλεγχο στον εκτελέσιμο κώδικα του boot sector του ενεργού διαμερίσματος.

- Εάν για κάποιο λόγο δεν καταφέρει να εκτελέσει τις παραπάνω ενέργειες εμφανίζει κάποιο από τα προκαθορισμένα μηνύματα σφάλματος (Invalid partition table, Error loading operating system, Missing operating system)

Εκόνα 2 :Ολόκληρο το MBR σε Hex και Ascii όπου διακρίνονται τα μηνύματα λάθους και η δομή του

Hard disk 2

Partitioning style: MBR

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
Partition 1	NTFS	90,9 GB					16128
Partition 2	NTFS	142 GB					190595160
Unpartitionable space		2,5 MB					488392065
Unpartitioned space		7,9 MB					0

[unregistriert]
Hard disk 2
Model: WDC WD2500JB-57REA0
Serial No.: WD-WCANK6150817
Firmware Rev.: 20.00K20
Bus: ATA

Default Edit Mode
State: original

Undo level: 0
Undo reverses: n/a

Total capacity: 233 GB
250.059.350.016 bytes

Bytes per sector: 512
Surplus sectors at end: 5103

Partition: <1
Relative sector No.: n/a

Mode: hexadecimal
Character set: ANSI ASCII
Offsets: hexadecimal
Bytes per page: 34x16=544

Window #: 1
No. of windows: 1

Clipboard: empty

TEMP folder: 7,4 GB free
CUME~1\cygnusx1\LOCALS~1\Temp

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00
0000000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00
0000000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10
0000000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00
0000000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09
0000000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74
0000000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00
0000000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13
0000000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00
0000000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE
00000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84
00000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55
00000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64
00000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75
00000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54
00000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00
0000000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66
0000000110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66
0000000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD
0000000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4
0000000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD
0000000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8
0000000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69
0000000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72
0000000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69
0000000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E
00000001A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74
00000001B0	65	6D	00	00	00	63	7B	9A	64	E1	64	E1	00	00	00	00
00000001C0	01	01	0F	FE	FF	FE	C1	3E	00	00	97	01	5C	0B	80	FE
00000001D0	FF	FE	07	FE	FF	FE	58	40	5C	0B	29	05	C0	11	00	00
00000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA
0000000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Ας ρίξουμε τώρα μια ματιά στα ενδότερα του εκτελέσιμου κώδικα για να δούμε τι ακριβώς κάνει:

Disassembly of the MBR

Ο τομέας αρχικά φορτώνεται στη διεύθυνση μνήμης 0000:7c00 αλλά αμέσως επανατοποθετείται στη 0000:0600.

```
                BEGIN:                          NOW AT 0000:7C00, RELOCATE

0000:7C00 FA            CLI                     disable int's
0000:7C01 33C0          XOR     AX,AX           set stack seg to 0000
0000:7C03 8ED0          MOV     SS,AX
0000:7C05 BC007C        MOV     SP,7C00         set stack ptr to 7c00
0000:7C08 8BF4          MOV     SI,SP           SI now 7c00
0000:7C0A 50            PUSH    AX
0000:7C0B 07            POP     ES              ES now 0000:7c00
0000:7C0C 50            PUSH    AX
0000:7C0D 1F            POP     DS              DS now 0000:7c00
0000:7C0E FB            STI                     allow int's
0000:7C0F FC            CLD                     clear direction
0000:7C10 BF0006        MOV     DI,0600         DI now 0600
0000:7C13 B90001        MOV     CX,0100         move 256 words (512 bytes)
0000:7C16 F2            REPNZ                   move MBR from 0000:7c00
0000:7C17 A5            MOVSW                      to 0000:0600
0000:7C18 EA1D060000    JMP     0000:061D       jmp to NEW_LOCATION

        NEW_LOCATION:                           NOW AT 0000:0600
Φόρτωσε στον SI τη διεύθυνση (07BE) όπου ξεκινούν οι εγγραφές για τα partition
0000:061D BEBE07        MOV     SI,07BE         point to first table entry
Που είναι 4
0000:0620 B304          MOV     BL,04           there are 4 table entries

        SEARCH_LOOP1:                           SEARCH FOR AN ACTIVE ENTRY
Σύγκρινε το πρώτο byte της πρώτης εγγραφής με το 80( που είναι η υπογραφή του ενεργού
διαμερίσματος)
0000:0622 803C80        CMP     BYTE PTR [SI],80  is this the active entry?
Εάν είναι ίσα «πήδα» το διαμέρισμα είναι ενεργό(active)
0000:0625 740E          JZ      FOUND_ACTIVE    yes
Διαφορετικά σύγρινέ το με το  00
0000:0627 803C00        CMP     BYTE PTR [SI],00  is this an inactive entry?
Εάν είναι ίσα δεν είναι ενεργό( αποδεκτή τιμή)/διαφορετικά «πήδα» "Invalid partition table"
0000:062A 751C          JNZ     NOT_ACTIVE      no
Συνέχισε με το επόμενο διαμέρισμα
0000:062C 83C610        ADD     SI,+10          incr table ptr by 16
0000:062F FECB          DEC     BL              decr count
0000:0631 75EF          JNZ     SEARCH_LOOP1    jmp if not end of table
Εάν δεν βρεθεί ενεργό διαμέρισμα καλείται η INT 18 και τυπώνεται κάποιο σχετικό μήνυμα
0000:0633 CD18          INT     18              GO TO ROM BASIC

        FOUND_ACTIVE:                           FOUND THE ACTIVE ENTRY

0000:0635 8B14          MOV     DX,[SI]         set DH/DL for INT 13 call
0000:0637 8B4C02        MOV     CX,[SI+02]      set CH/CL for INT 13 call
Αποθήκευσε το δείκτη του ενεργού διαμερίσματος
0000:063A 8BEE          MOV     BP,SI           save table ptr

        SEARCH_LOOP2:                           MAKE SURE ONLY ONE ACTIVE ENTRY
Έλεγξε εάν το ενεργό διαμέρισμα που βρήκες είναι και το μοναδικό
Κοίταξε στην επόμενη εγγραφή που είναι 16 bytes μετά την πρώτη
0000:063C 83C610        ADD     SI,+10          incr table ptr by 16
Απομένουν για έλεγχο άλλα … 3,2,1 διαμερίσματα
0000:063F FECB          DEC     BL              decr count
Εάν τα έλεγξες όλα «πήδα»
0000:0641 741A          JZ      READ_BOOT       jmp if end of table
0000:0643 803C00        CMP     BYTE PTR [SI],00  is this an inactive entry?
```

```
0000:0646 74F4        JZ      SEARCH_LOOP2      yes
 εάν  έχουμε περισσότερα από 1 ενεργά διαμερίσματα ή κάποια άκυρη εγγραφή (ούτε 80 ούτε 00)
         NOT_ACTIVE:                          MORE THAN ONE ACTIVE ENTRY FOUND

0000:0648 BE8B06      MOV     SI,068B           display "Invld prttn tbl"

        DISPLAY_MSG:                          DISPLAY MESSAGE LOOP

0000:064B AC          LODSB                     get char of message
0000:064C 3C00        CMP     AL,00             end of message
0000:064E 740B        JZ      HANG              yes
0000:0650 56          PUSH    SI                save SI
0000:0651 BB0700      MOV     BX,0007           screen attributes
0000:0654 B40E        MOV     AH,0E             output 1 char of message
0000:0656 CD10        INT     10                   to the display
0000:0658 5E          POP     SI                restore SI
0000:0659 EBF0        JMP     DISPLAY_MSG       do it again

               HANG:                          HANG THE SYSTEM LOOP

0000:065B EBFE        JMP     HANG              sit and stay!

Διάβασε την εγγραφή εκκίνησης του ενεργού διαμερίσματος
          READ_BOOT:                          READ ACTIVE PARITION BOOT RECORD
Μέτρα 5 προσπάθειες
0000:065D BF0500      MOV     DI,0005           INT 13 retry count

          INT13RTRY:                          INT 13 RETRY LOOP

0000:0660 BB007C      MOV     BX,7C00
Διαβασε(AH=02h) τον πρώτο(AH=01h) τομέα του ενεργού διαμερίσματος(boot record)
0000:0663 B80102      MOV     AX,0201           read 1 sector
0000:0666 57          PUSH    DI                save DI
Φόρτωσέ τον στη θέση μνήμης 0000:7c00
0000:0667 CD13        INT     13                read sector into 0000:7c00
0000:0669 5F          POP     DI                restore DI
Εάν πετύχει η INT13 (CF=0) «πήδα»
0000:066A 730C        JNB     INT13OK           jmp if no INT 13
Διαφορετικά προσπάθησε πάλι
0000:066C 33C0        XOR     AX,AX             call INT 13 and
0000:066E CD13        INT     13                   do disk reset
0000:0670 4F          DEC     DI                decr DI
Αν αποτύχεις και την 5η φορά
0000:0671 75ED        JNZ     INT13RTRY         if not zero, try again
0000:0673 BEA306      MOV     SI,06A3           display "Errr ldng systm"
Τύπωσε το μήνυμα :Error loading system
0000:0676 EBD3        JMP     DISPLAY_MSG       jmp to display loop

            INT13OK:                          INT 13 ERROR

Ετοίμασε το μήνυμα :Missing operating system
0000:0678 BEC206      MOV     SI,06C2             "missing op sys"
0000:067B BFFE7D      MOV     DI,7DFE             point to signature
Διάβασε την υπογραφή του τομέα
0000:067E 813D55AA    CMP     WORD PTR [DI],AA55  is signature correct?
Εάν είναι σωστή (AA55) συνέχισε/αλλιώς τύπωσε το μήνυμα
0000:0682 75C7        JNZ     DISPLAY_MSG         no
0000:0684 8BF5        MOV     SI,BP               set SI
Και πήγαινε στη θέση μνήμης που φόρτωσες προηγουμένως το Boot Sector και δώσε τον έλεγχο
0000:0686 EA007C0000  JMP     0000:7C00           JUMP TO THE BOOT SECTOR
                                                     WITH SI POINTING TO
                                                     PART TABLE ENTRY

Messages here.

0000:0680 ........ ........ ......49 6e76616c *           Inval*
0000:0690 69642070 61727469 74696f6e 20746162 *id partition tab*
0000:06a0 6c650045 72726f72 206c6f61 64696e67 *le.Error loading*
```

```
0000:06b0 206f7065 72617469 6e672073 79737465 * operating syste*
0000:06c0 6d004d69 7373696e 67206f70 65726174 *m.Missing operat*
0000:06d0 696e6720 73797374 656d00.. ........ *ing system.     *

Data not used.

0000:06d0 ........ ........ ......00 00000000 *           .....*
0000:06e0 00000000 00000000 00000000 00000000 *................*
0000:06f0 00000000 00000000 00000000 00000000 *................*
0000:0700 00000000 00000000 00000000 00000000 *................*
0000:0710 00000000 00000000 00000000 00000000 *................*
0000:0720 00000000 00000000 00000000 00000000 *................*
0000:0730 00000000 00000000 00000000 00000000 *................*
0000:0740 00000000 00000000 00000000 00000000 *................*
0000:0750 00000000 00000000 00000000 00000000 *................*
0000:0760 00000000 00000000 00000000 00000000 *................*
0000:0770 00000000 00000000 00000000 00000000 *................*
0000:0780 00000000 00000000 00000000 00000000 *................*
0000:0790 00000000 00000000 00000000 00000000 *................*
0000:07a0 00000000 00000000 00000000 00000000 *................*
0000:07b0 00000000 00000000 00000000 0000.... *............    *

The partition table starts at 0000:07be.  Each partition table
entry is 16 bytes.  This table defines a single primary partition
which is also an active (bootable) partition.

0000:07b0 ........ ........ ........ ....8001 *            ....*
0000:07c0 0100060d fef83e00 00000678 0d000000 *..........x....*
0000:07d0 00000000 00000000 00000000 00000000 *................*
0000:07e0 00000000 00000000 00000000 00000000 *................*
0000:07f0 00000000 00000000 00000000 0000.... *............    *

The last two bytes contain a 55AAH signature.

0000:07f0 ........ ........ ........ ....55aa *..............U.*
```

Συνοψίζοντας λοιπόν θα λέγαμε ότι προβλήματα μπορεί να υπάρξουν σ΄ αυτό το στάδιο της εκκίνησης αν:

- Δε βρεθεί ενεργό διαμέρισμα ---> το μήνυμα εξαρτάται από το BIOS
- Βρεθούν περισσότερα από 1 ενεργά διαμερίσματα ή κάποια άκυρη εγγραφή στο αντίστοιχο πεδίο Table primary partition(το πρώτο byte σε κάθε εγγραφή) όπου αναμένεται μια τιμή 0x80(active) ή 0x00 (no active) -----> “ invalid partition table”
- Δεν καταφέρει να φορτώσει τον PBR (partition boot record) στη μνήμη -----> Error loading operating system
- Δεν εντοπίσει την υπογραφή (0x55AA)στο τέλος του PBR (δεν υπάρχει boot sector)---> Missing operating system

Ας δούμε τώρα και τη δομή του Table of primary partition:

MBRs and disk partitioning

The MBR is not located in a partition, it is located at a Main Boot Record area in front of the first partition.

When a data storage device has been partitioned with the MBR Partition Table scheme (i.e., the conventional IBM PC partitioning scheme), the master boot record contains the primary partition entries in its partition table. The partition table may also contain entries for other, secondary partitions which are stored in extended boot records (EBRs), BSD disklabels, and Logical Disk Manager metadata partitions that are described by those primary entries.[15]

Layout of one 16-byte partition record

Offset	Field length (bytes)	Description
0x00	1	status[7] (0x80 = bootable (*active*), 0x00 = non-bootable, other = invalid[8])
0x01	3	CHS address of first block in partition.[9] The format is described in the next 3 bytes.
0x01	1	head[10]
0x02	1	sector is in bits 5–0[11]; bits 9–8 of cylinder are in bits 7–6
0x03	1	bits 7–0 of cylinder[12]
0x04	1	partition type[13]
0x05	3	CHS address of last block in partition.[14] The format is described in the next 3 bytes.
0x05	1	head
0x06	1	sector is in bits 5–0; bits 9–8 of cylinder are in bits 7–6
0x07	1	bits 7–0 of cylinder
0x08	4	LBA of first sector in the partition
0x0C	4	number of blocks in partition, in little-endian format

By convention, there are exactly four primary partition table entries in the MBR Partition Table scheme, although some DOS operating systems did extend this to five (PTS-DOS)[16] or even eight (AST or NEC DOS)[17][18] entries. Both the partition length and partition start address are stored as 32-bit quantities. Because the block size is 512 bytes, this implies that neither the maximum size of a partition nor the maximum start address (both in bytes) can exceed $2^{32} \times 512$ bytes, or 2 TiB. Alleviating this capacity limitation is one of the prime motivations for the development of the GUID Partition Table (GPT).

Where a data storage device has been partitioned with the GPT scheme, the Master Boot Record will still contain a partition table, but its only purpose is to indicate the existence of the GUID Table and to prevent utility programs that understand only the MBR Partition Table scheme from creating any partitions in what they would see as free space on the disk, thereby accidentally erasing the GUID table.

1.Το πρώτο(1^o) byte χρησιμοποιείται για τη δήλωση του ενεργού διαμερίσματος:

- 0x80 ενεργό –εκκίνηση από αυτό
- 0x00 μη ενεργό- μη εκκινήσιμο
- Άλλο άκυρο – διακοπή διαδικασίας εκκίνησης

2.Τα επόμενα 3(2^o -3^o -4^o) bytes μας δίνουν τη διεύθυνση του πρώτου block (τη μικρότερη κατάτμηση του σκληρού δίσκου, που αντιπροσωπεύει την ελάχιστη ποσότητα δεδομένων που μπορεί να γραφτεί ή να διαβαστεί από το δίσκο. Συνήθως αναφέρεται και ως sector που όμως δεν είναι το ίδιο) σε μορφή CHS (Cylinder-Head-Sector).

3.Το 5ο byte προσδιορίζει τον τύπο του διαμερίσματος (π.χ.NTFS)

4.Τα άλλα 3 bytes (6^o-7^o-8^o) μας δίνουν τη διεύθυνση του τελευταίου block σε μορφή CHS

5. Ακολουθούν 4 bytes (9^o -10^o -11^o -12^o) που δείχνουν την LBA (logicalBlockAddressing στην ουσία πρόκειται για ένα γραμμικό τρόπο αναφοράς στα blocks του σκληρού με χρήση δεικτών π.χ.LBA=0 για το πρώτο ,LBA=1 για το δεύτερο κ.ο.κ.) του πρώτου block στο partition.
6.Τα τελευταία 4 bytes (13^o -14^o -15^o-16^o) είναι ο συνολικός αριθμός των blocks του διαμερίσματος, που αν τον πολλαπλασιάσουμε με το μέγεθος του block (συνήθως 512 bytes) βρίσκουμε τη συνολική χωρητικότητα του διαμερίσματος.
Βέβαια ο partition table μπορεί να περιέχει πληροφορίες για άλλα λογικά διαμερίσματα στο δίσκο (extended pertition)

**********Minde the little-endian format**********

Νικολόπουλος Δημήτρης