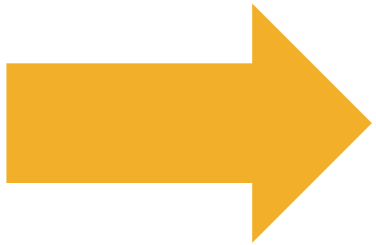
Einführung in die Betriebssysteme

Martin Spörl

Bootvorgang

Grundlagen

- Computer besteht aus
  - Rechenwerk
  - Primärspeicher (flüchtig)
  - Sekundärspeicher (nicht flüchtig)
- OS liegt oft auf Sekundärspeicher
- Sekundärspeicher oft „Add-on“
- Prozessor muss herausfinden, wie er anfangen soll



Booten eines Computers

- mehrstufiges Starten des Computers bis zum OS
- prüfen der Hardware auf Fehler
- Starten / Laden des OS

Wortherkunft

“to pull oneself up by one's bootstraps”

- englisches Sprichwort als Grundlagen
- Computer lädt Programm, dass es ihm ermöglicht dass er funktioniert
- Computer zieht sich sozusagen selbst an den Haaren hoch (wie Baron von Münchhausen)
- Im Engl. Zieht man sich an den Stiefelschlaufen („bootstraps“)

Booting

Herkömmliches Booten

Kaltstart

- Stromzufuhr war unterbrochen
- Computer muss „von 0 starten“.
- Auslöser:
  - Computer einschalten
  - Reset-Taste gedrückt

Warmstart

- Stromzufuhr nicht unterbrochen
- Oft wird Hardware Initiierung nicht erneut ausgeführt
- Je nach OS / Rechnerarchitektur mehr oder weniger anders als Kaltstart
  - x86: Komplett Neustart – alle flüchtigen Daten weg
- Auslöser:
  - Neustart durch das OS (z.B. Windows > Neustart

Energiesparfunktionen

Suspend to disk (Ruhemodus)

- Idee: Computer wird in Stromlosen Zustand versetzt
- RAM wird auf Festplatte gespeichert
- Peripherie wird abgeschaltet
- Bei Start: RAM-Abbild wird von Festplatte in den Speicher geladen

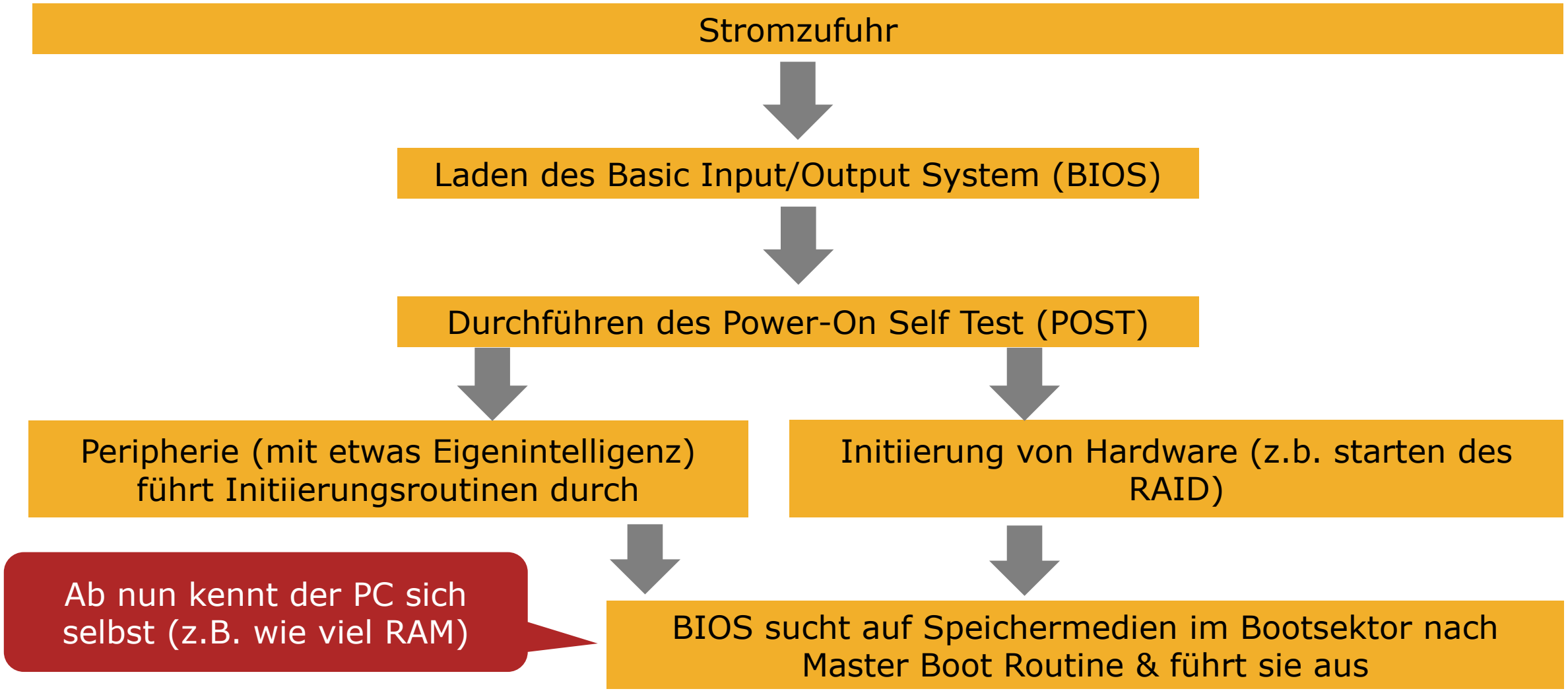
Suspend to ram (Standby Modus)

- Idee: große Stromverbraucher abschalten
- Deaktivieren von nicht gebrauchter Peripherie
  - Bildschirm
  - Festplatte
  - CD Laufwerk
- Prozessor und RAM bleiben weitestgehend aktiv

Suspend to ram and disk

- Mischung aus Ruhemodus und Standby Modus
- RAM wird auf Festplatte gespeichert
- Festplatte wird abgeschaltet
- RAM-Abbild wird nur dann verwendet wenn Stromversorgung unterbrochen war (z.b. Akku leer)
- Unterschiedliche Namen in OS
  - Windows: Hibernate („Energiesparen“)
  - Mac OS: Safe Sleep („Sicherer Ruhezustand)

Ablauf – Allgemeiner Start

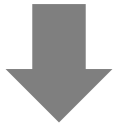


Ablauf – Windows (vereinfacht)

BIOS startet Windows Bootloader (NTLDR (bis XP) / BOOTMGR (ab Vista))

Bootloader startet Winload.exe

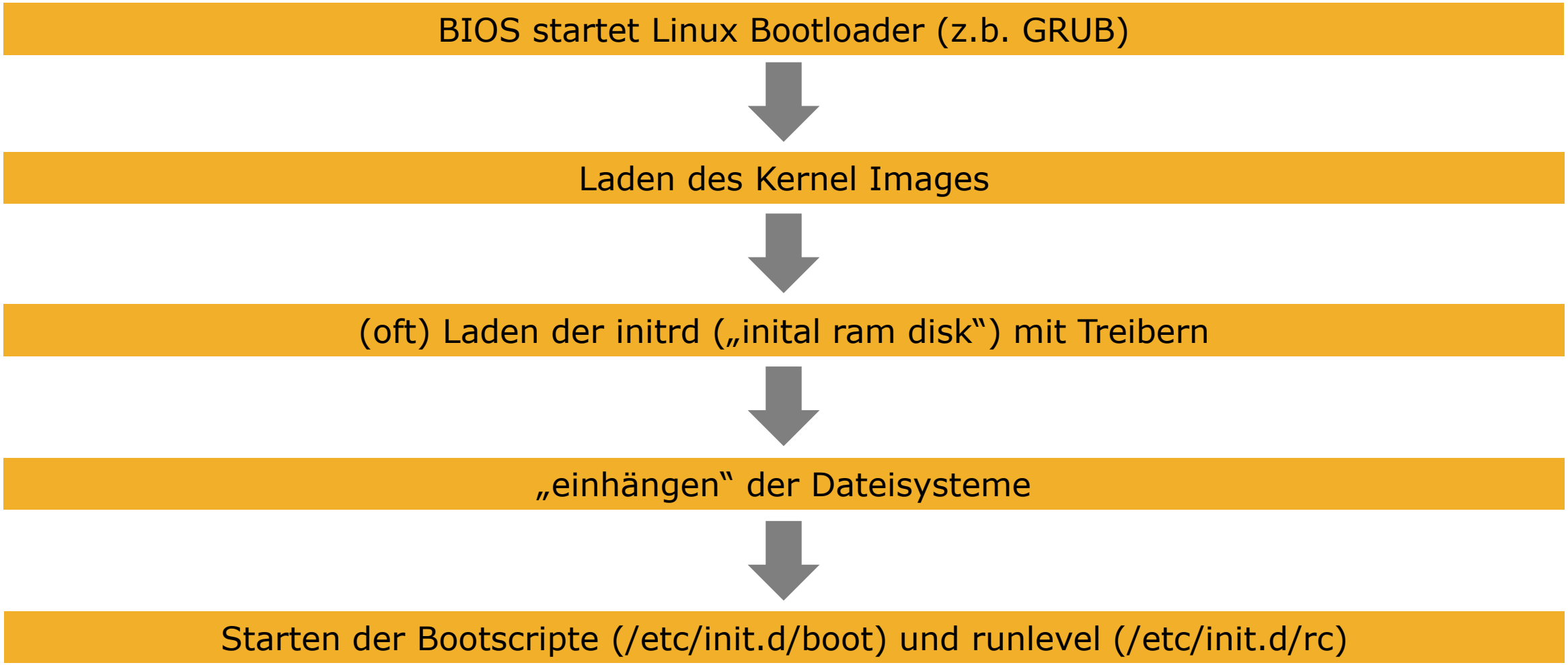
Winload.exe lädt treiber (um Hardware Abstraction Layer (HAL) aufzubauen)



Winload.exe lädt Kernel & Registry

Übergabe der Kontrolle an den Kernel

Ablauf – Linux (vereinfacht)



BIOS vs. UEFI

BIOS (Basic Input/Output System)

- Historischer Hintergrund
- Früher: wenig Speicher
  - Bootprozess wurde aufgeteilt und nach und nach geladen um Platz zu sparen
- Wird auf ROM auf dem Motherboard gespeichert
- Aufgaben
  - Durchführen des POST
  - Initiierung der Peripherie
  - Starten des OS

UEFI (Unified Extensible Firmware Interface)

- Nachfolger von BIOS
- Implementiert viele bei BIOS vermissten Punkte
  - DRM (Digital Rights Management)
  - PXE (bisher nur via Netzwerkkarte)
  - Hochauflösende Grafik bei Boot
  - Eigenen Shell
  - GPT (GUID Partition Table) soll flexibler als MBR sein)
- Stößt auf Widerstand bei BIOS / Mainboard und OS Herstellern
- Bisherige Unterstützer
  - Apple (seit umstieg auf x86)
  - Intel
  - Microsoft (mit Abstrichen)

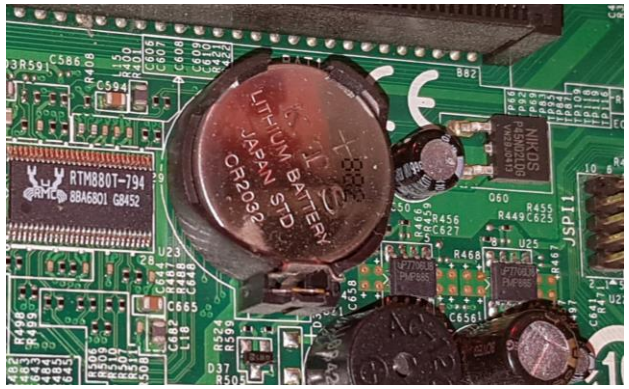
„BIOS - Speicher“

CMOS Chip

- „CMOS“ Bezeichnet eigentlich den Herstellungsprozess
- „Complementary Metal-Oxide-Semiconductor“
- Ist ein RAM (eigentlich flüchtig)
  - braucht daher eigene Stromversorgung
- Heute oft mit Echtzeituhr in einem Chip
  - beide benötigen durchgehende Stromversorgung
- Speichert alle BIOS Einstellungen
  - Boot Order
  - Virtualisierungseinstellung
  - Uhrzeit / Datum
  - …
- Oft nur wenige Byte groß

CMOS-Batterie

- versorgt CMOS-Chip & Echtzeituhr mit Strom, auch wenn PC vom Strom getrennt
- häufigste Batterie: CR2032 Lithium Batterie
  - Früher kamen auch AA-Batterien zum Einsatz
- häufigster Indikator dass Batterie leer ist
  - OS startet mit falschen Datum (z.B. 1.1.1970)
  - alle BIOS Einstellung auf Werkszustand



POST I

Power-On Self Test

- Sorgt für fehlerfreie Hardware
  - essentiell für fehlerfreie Programm / OS Ausführung
- Erstes Programm dass vom BIOS gestartet wird
- Prozessor prüft sich selbst und angeschlossene Peripherie
- Kann via Monitor oder Biep Codes Rückmeldung geben
- Ausgelöst durch elektrisches Signal beim Starten an den Prozessor

Biep Codes

- Von Hersteller zu Hersteller unterschiedlich
- Weiche oft sogar von BIOS Version zu Version ab
- Sind eine Art Morsecode
- Ermögliche Rückmeldung auch ohne Monitor

Beispiel von HP Rechnern

Beeps	Description
1 short beep and 1 long beep	Memory problem
2 short beeps and 1 long beep (repeats 5 times)	Unable to initialize video or video card required but not installed
3 short beeps and 1 long beep	CPU configuration error or CPU type is not compatible

POST II - Ablauf

1. Prozessor prüft sich selbst
2. Prozessor sendet Signal über Systembus (prüfen ob Komponenten funktionieren)
3. Prozessor prüft Systemuhr
4. Prozessor prüft Grafikkarte

Ab nun Ausgabe auf Monitor

5. Arbeitsspeicher wird geprüft
6. Prozessor prüft Tastatur (z.B. ob vorhanden)
7. Prozessor prüft vorhandene Laufwerke
8. Falls neue Komponenten gefunden werden wird Konfigurationsroutine gestartet
9. Komponenten mit eigenen Selbsttest werden einbezogen (z.B. „SMART“ – Test bei Festplatten)
10. Fortführung des booten

MBR

Master Boot Record

- erste 512 Byte auf der Festplatte
- ist in 3 Teile geteilt
  - Master Boot Routine
  - Partitionstabelle
  - *Magic Number*

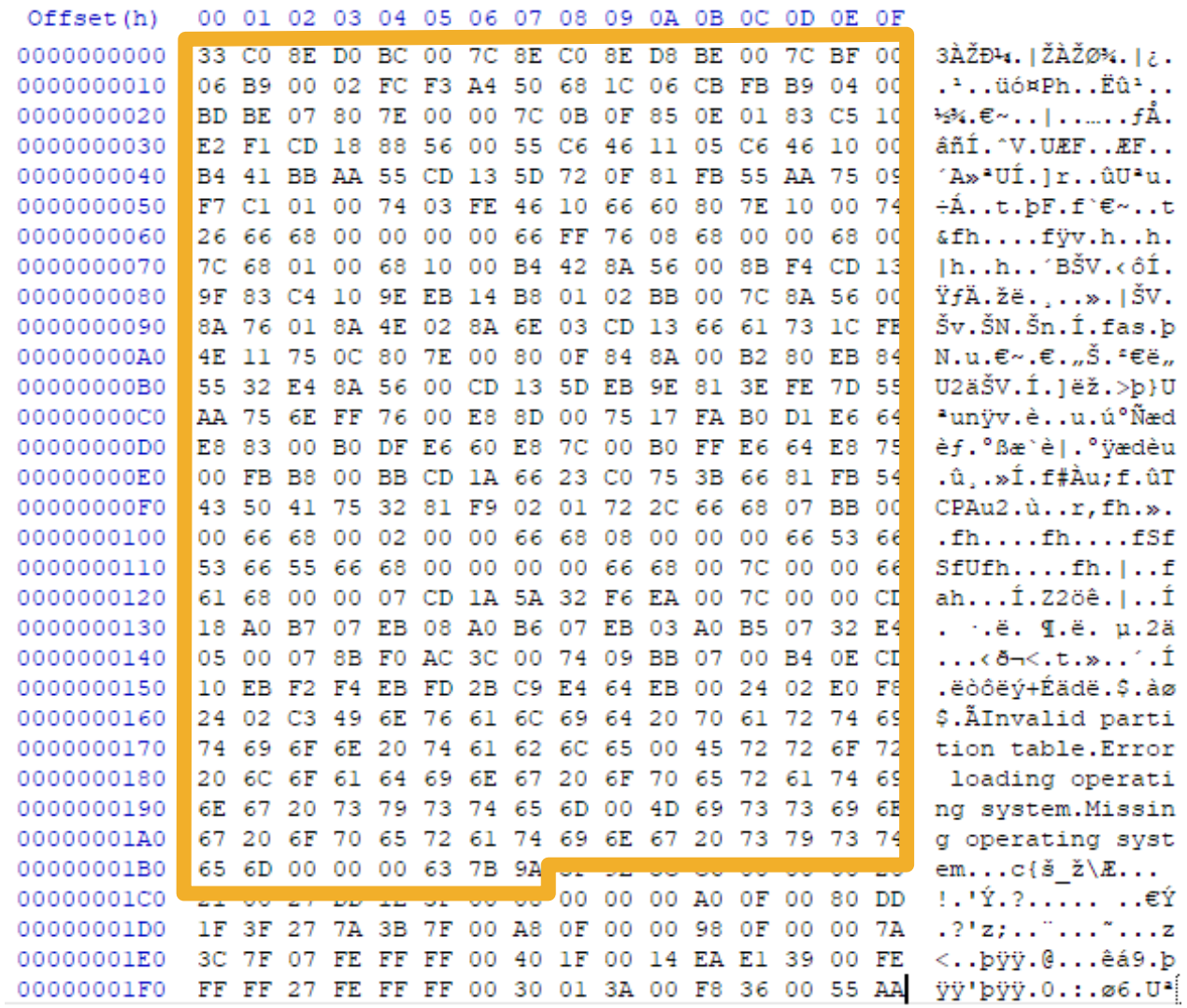
ist die Startroutine für das Betriebssystem

Volume Boot Record

- wie MBR, nur die ersten 512 Byte der Partition
- andere Aufteilung
- abhängig vom Dateisystem

Adresse	Inhalt	Länge (Byte)
0x0000	Master Boot Routine	440
0x01B8	Datenträgersignatur (seit Windows 2000)	4
0x01BC	0x0000	2
0x01BE	Partitionstabelle	64
0x01FE 0x01FF	Magic Number („Bootsector Signature“ – immer x55AA)	2

MBR - Beispiel



```
Offset(h)   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00  3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
0000000010  06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB B9 04 00  .¹..üó¤Ph..ËÛ¹..
0000000020  BD BE 07 80 7E 00 00 7C 0B 0F 85 0E 01 83 C5 10  ½¾.€~..|.....ƒÅ.
0000000030  E2 F1 CD 18 88 56 00 55 C6 46 11 05 C6 46 10 00  âñÍ.ˆV.UÆF..ÆF..
0000000040  B4 41 BB AA 55 CD 13 5D 72 0F 81 FB 55 AA 75 09  ´A»ªUÍ.]r..ûUªu.
0000000050  F7 C1 01 00 74 03 FE 46 10 66 60 80 7E 10 00 74  ÷Á..t.þF.f`€~..t
0000000060  26 66 68 00 00 00 00 66 FF 76 08 68 00 00 68 00  &fh....fÿv.h..h.
0000000070  7C 68 01 00 68 10 00 B4 42 8A 56 00 8B F4 CD 13  |h..h..´BŠV.‹ôÍ.
0000000080  9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00  ŸƒÄ.žë.¸..».|ŠV.
0000000090  8A 76 01 8A 4E 02 8A 6E 03 CD 13 66 61 73 1C FE  Šv.ŠN.Šn.Í.fas.þ
00000000A0  4E 11 75 0C 80 7E 00 80 0F 84 8A 00 B2 80 EB 84  N.u.€~.€.„Š.²€ë„
00000000B0  55 32 E4 8A 56 00 CD 13 5D EB 9E 81 3E FE 7D 55  U2äŠV.Í.]ëž.>þ}U
00000000C0  AA 75 6E FF 76 00 E8 8D 00 75 17 FA B0 D1 E6 64  ªunÿv.è..u.ú°Ñæd
00000000D0  E8 83 00 B0 DF E6 60 E8 7C 00 B0 FF E6 64 E8 75  èƒ.°ßæ`è|.°ÿædèu
00000000E0  00 FB B8 00 BB CD 1A 66 23 C0 75 3B 66 81 FB 54  .û¸.»Í.f#Àu;f.ûT
00000000F0  43 50 41 75 32 81 F9 02 01 72 2C 66 68 07 BB 00  CPAu2.ù..r,fh.».
0000000100  00 66 68 00 02 00 00 66 68 08 00 00 00 66 53 66  .fh....fh....fSf
0000000110  53 66 55 66 68 00 00 00 00 66 68 00 7C 00 00 66  SfUfh....fh.|..f
0000000120  61 68 00 00 07 CD 1A 5A 32 F6 EA 00 7C 00 00 CD  ah...Í.Z2öê.|..Í
0000000130  18 A0 B7 07 EB 08 A0 B6 07 EB 03 A0 B5 07 32 E4  . ·.ë. ¶.ë. µ.2ä
0000000140  05 00 07 8B F0 AC 3C 00 74 09 BB 07 00 B4 0E CD  ...‹ð¬<.t.»..´.Í
0000000150  10 EB F2 F4 EB FD 2B C9 E4 64 EB 00 24 02 E0 F8  .ëòôëý+Éädë.$.àø
0000000160  24 02 C3 49 6E 76 61 6C 69 64 20 70 61 72 74 69  $.ÃInvalid parti
0000000170  74 69 6F 6E 20 74 61 62 6C 65 00 45 72 72 6F 72  tion table.Error
0000000180  20 6C 6F 61 64 69 6E 67 20 6F 70 65 72 61 74 69   loading operati
0000000190  6E 67 20 73 79 73 74 65 6D 00 4D 69 73 73 69 6E  ng system.Missin
00000001A0  67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
00000001B0  65 6D 00 00 00 63 7B 9A [illegible]              em...c{š_ž\Æ...
00000001C0  [illegible] 00 00 00 A0 0F 00 80 DD              !.'Ý.?..... ..€Ý
00000001D0  1F 3F 27 7A 3B 7F 00 A8 0F 00 00 98 0F 00 00 7A  .?'z;..¨...˜...z
00000001E0  3C 7F 07 FE FF FF 00 40 1F 00 14 EA E1 39 00 FE  <..þÿÿ.@...êá9.þ
00000001F0  FF FF 27 FE FF FF 00 30 01 3A 00 F8 36 00 55 AA  ÿÿ'þÿÿ.0.:.ø6.Uª
```

Master Boot Routine

MBR - Beispiel

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

0000000000  33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00  3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
0000000010  06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB B9 04 00  .¹..üó¤Ph..ËÛ¹..
0000000020  BD BE 07 80 7E 00 00 7C 0B 0F 85 0E 01 83 C5 10  ½¾.€~..|.....ƒÅ.
0000000030  E2 F1 CD 18 88 56 00 55 C6 46 11 05 C6 46 10 00  âñÍ.ˆV.UÆF..ÆF..
0000000040  B4 41 BB AA 55 CD 13 5D 72 0F 81 FB 55 AA 75 09  ´A»ªUÍ.]r..ûUªu.
0000000050  F7 C1 01 00 74 03 FE 46 10 66 60 80 7E 10 00 74  ÷Á..t.þF.f`€~..t
0000000060  26 66 68 00 00 00 00 66 FF 76 08 68 00 00 68 00  &fh....fÿv.h..h.
0000000070  7C 68 01 00 68 10 00 B4 42 8A 56 00 8B F4 CD 13  |h..h..´BŠV.‹ôÍ.
0000000080  9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00  ŸƒÄ.žë.¸..».|ŠV.
0000000090  8A 76 01 8A 4E 02 8A 6E 03 CD 13 66 61 73 1C FE  Šv.ŠN.Šn.Í.fas.þ
00000000A0  4E 11 75 0C 80 7E 00 80 0F 84 8A 00 B2 80 EB 84  N.u.€~.€.„Š.²€ë„
00000000B0  55 32 E4 8A 56 00 CD 13 5D EB 9E 81 3E FE 7D 55  U2äŠV.Í.]ëž.>þ}U
00000000C0  AA 75 6E FF 76 00 E8 8D 00 75 17 FA B0 D1 E6 64  ªunÿv.è..u.ú°Ñæd
00000000D0  E8 83 00 B0 DF E6 60 E8 7C 00 B0 FF E6 64 E8 75  èƒ.°ßæ`è|.°ÿædèu
00000000E0  00 FB B8 00 BB CD 1A 66 23 C0 75 3B 66 81 FB 54  .û¸.»Í.f#Àu;f.ûT
00000000F0  43 50 41 75 32 81 F9 02 01 72 2C 66 68 07 BB 00  CPAu2.ù..r,fh.».
0000000100  00 66 68 00 02 00 00 66 68 08 00 00 00 66 53 66  .fh....fh....fSf
0000000110  53 66 55 66 68 00 00 00 00 66 68 00 7C 00 00 66  SfUfh....fh.|..f
0000000120  61 68 00 00 07 CD 1A 5A 32 F6 EA 00 7C 00 00 CD  ah...Í.Z2öê.|..Í
0000000130  18 A0 B7 07 EB 08 A0 B6 07 EB 03 A0 B5 07 32 E4  . ·.ë. ¶.ë. µ.2ä
0000000140  05 00 07 8B F0 AC 3C 00 74 09 BB 07 00 B4 0E CD  ...‹ð¬<.t.»..´.Í
0000000150  10 EB F2 F4 EB FD 2B C9 E4 64 EB 00 24 02 E0 F8  .ëòôëý+Éädë.$.àø
0000000160  24 02 C3 49 6E 76 61 6C 69 64 20 70 61 72 74 69  $.ÃInvalid parti
0000000170  74 69 6F 6E 20 74 61 62 6C 65 00 45 72 72 6F 72  tion table.Error
0000000180  20 6C 6F 61 64 69 6E 67 20 6F 70 65 72 61 74 69   loading operati
0000000190  6E 67 20 73 79 73 74 65 6D 00 4D 69 73 73 69 6E  ng system.Missin
00000001A0  67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
00000001B0  65 6D 00 00 00 63 7B 9A 5F 9E 5C C6 00 00 00 20  em...c{š_ž\Æ...
00000001C0  21 00 27 DD 1E 3F 00 08 00 00 A0 0F 00 80 DD  !.'Ý.?..... ..€Ý
00000001D0  1F 3F 27 7A 3B 7F 00 A8 0F 00 00 98 0F 00 00 7A  .?'z;..¨...˜...z
00000001E0  3C 7F 07 FE FF FF 00 40 1F 00 14 EA E1 39 00 FE  <..þÿÿ.@...êá9.þ
00000001F0  FF FF 27 FE FF FF 00 30 01 3A 00 F8 36 00 55 AA  ÿÿ'þÿÿ.0.:.ø6.Uª
```

Datenträgersignatur
(seit Windows 2000)

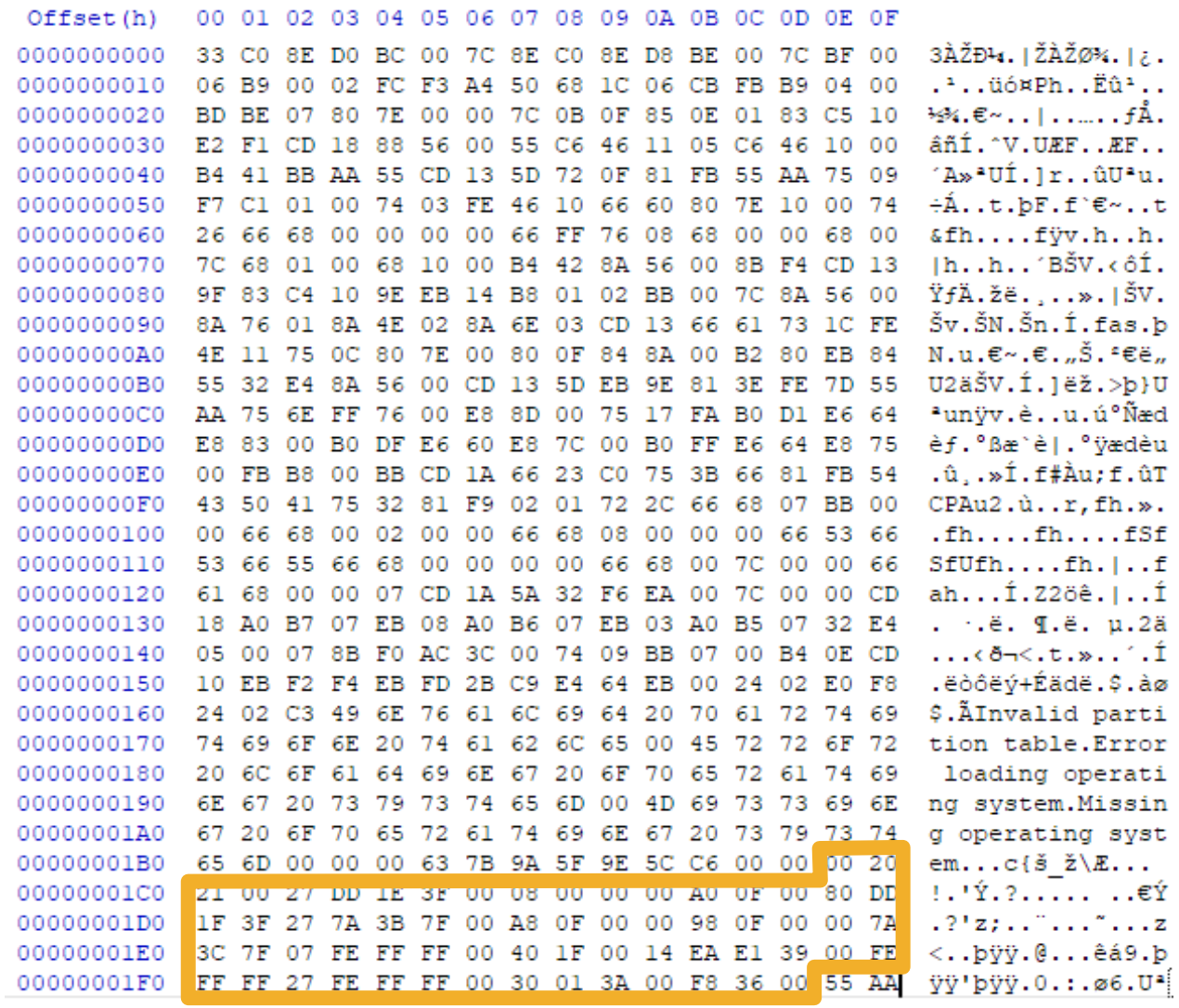
MBR - Beispiel

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

0000000000  33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00  3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
0000000010  06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB B9 04 00  .¹..üó¤Ph..ËÛ¹..
0000000020  BD BE 07 80 7E 00 00 7C 0B 0F 85 0E 01 83 C5 10  ½¾.€~..|.....ƒÅ.
0000000030  E2 F1 CD 18 88 56 00 55 C6 46 11 05 C6 46 10 00  âñÍ.ˆV.UÆF..ÆF..
0000000040  B4 41 BB AA 55 CD 13 5D 72 0F 81 FB 55 AA 75 09  ´A»ªUÍ.]r..ûUªu.
0000000050  F7 C1 01 00 74 03 FE 46 10 66 60 80 7E 10 00 74  ÷Á..t.þF.f`€~..t
0000000060  26 66 68 00 00 00 00 66 FF 76 08 68 00 00 68 00  &fh....fÿv.h..h.
0000000070  7C 68 01 00 68 10 00 B4 42 8A 56 00 8B F4 CD 13  |h..h..´BŠV.‹ôÍ.
0000000080  9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00  ŸƒÄ.žë.¸..».|ŠV.
0000000090  8A 76 01 8A 4E 02 8A 6E 03 CD 13 66 61 73 1C FE  Šv.ŠN.Šn.Í.fas.þ
00000000A0  4E 11 75 0C 80 7E 00 80 0F 84 8A 00 B2 80 EB 84  N.u.€~.€.„Š.²€ë„
00000000B0  55 32 E4 8A 56 00 CD 13 5D EB 9E 81 3E FE 7D 55  U2äŠV.Í.]ëž.>þ}U
00000000C0  AA 75 6E FF 76 00 E8 8D 00 75 17 FA B0 D1 E6 64  ªunÿv.è..u.ú°Ñæd
00000000D0  E8 83 00 B0 DF E6 60 E8 7C 00 B0 FF E6 64 E8 75  èƒ.°ßæ`è|.°ÿædèu
00000000E0  00 FB B8 00 BB CD 1A 66 23 C0 75 3B 66 81 FB 54  .û¸.»Í.f#Àu;f.ûT
00000000F0  43 50 41 75 32 81 F9 02 01 72 2C 66 68 07 BB 00  CPAu2.ù..r,fh.».
0000000100  00 66 68 00 02 00 00 66 68 08 00 00 00 66 53 66  .fh....fh....fSf
0000000110  53 66 55 66 68 00 00 00 00 66 68 00 7C 00 00 66  SfUfh....fh.|..f
0000000120  61 68 00 00 07 CD 1A 5A 32 F6 EA 00 7C 00 00 CD  ah...Í.Z2öê.|..Í
0000000130  18 A0 B7 07 EB 08 A0 B6 07 EB 03 A0 B5 07 32 E4  . ·.ë. ¶.ë. µ.2ä
0000000140  05 00 07 8B F0 AC 3C 00 74 09 BB 07 00 B4 0E CD  ...‹ð¬<.t.»..´.Í
0000000150  10 EB F2 F4 EB FD 2B C9 E4 64 EB 00 24 02 E0 F8  .ëòôëý+Éädë.$.àø
0000000160  24 02 C3 49 6E 76 61 6C 69 64 20 70 61 72 74 69  $.ÃInvalid parti
0000000170  74 69 6F 6E 20 74 61 62 6C 65 00 45 72 72 6F 72  tion table.Error
0000000180  20 6C 6F 61 64 69 6E 67 20 6F 70 65 72 61 74 69   loading operati
0000000190  6E 67 20 73 79 73 74 65 6D 00 4D 69 73 73 69 6E  ng system.Missin
00000001A0  67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
00000001B0  65 6D 00 00 00 63 7B 9A 5F 9E 5C C6 00 00 00 20  em...c{š_ž\Æ...
00000001C0  21 00 27 DD 1E 3F 00 08 00 00 00 A0 01 00 80 DD  !.'Ý.?..... ..€Ý
00000001D0  1F 3F 27 7A 3B 7F 00 A8 0F 00 00 98 0F 00 00 7A  .?'z;..¨...˜...z
00000001E0  3C 7F 07 FE FF FF 00 40 1F 00 14 EA E1 39 00 FE  <..þÿÿ.@...êá9.þ
00000001F0  FF FF 27 FE FF FF 00 30 01 3A 00 F8 36 00 55 AA  ÿÿ'þÿÿ.0.:.ø6.Uª
```

Empty

MBR - Beispiel



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0000000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00	3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
0000000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00	.¹..üó¤Ph..ËÛ¹..
0000000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10	½¾.€~..|.....ƒÅ.
0000000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00	âñÍ.ˆV.UÆF..ÆF..
0000000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09	´A»ªUÍ.]r..ûUªu.
0000000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74	÷Á..t.þF.f`€~..t
0000000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00	&fh....fÿv.h..h.
0000000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13	|h..h..´BŠV.‹ôÍ.
0000000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00	ŸƒÄ.žë.¸..».|ŠV.
0000000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE	Šv.ŠN.Šn.Í.fas.þ
00000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84	N.u.€~.€.„Š.²€ë„
00000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55	U2äŠV.Í.]ëž.>þ}U
00000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64	ªunÿv.è..u.ú°Ñæd
00000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75	èƒ.°ßæ`è|.°ÿædèu
00000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54	.û¸.»Í.f#Àu;f.ûT
00000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00	CPAu2.ù..r,fh.».
0000000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66	.fh....fh....fSf
0000000110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66	SfUfh....fh.|..f
0000000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD	ah...Í.Z2öê.|..Í
0000000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4	. ·.ë. ¶.ë. µ.2ä
0000000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD	...‹ð¬<.t.»..´.Í
0000000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8	.ëòôëý+Éädë.$.àø
0000000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69	$.ÃInvalid parti
0000000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72	tion table.Error
0000000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69	loading operati
0000000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E	ng system.Missin
00000001A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
00000001B0	65	6D	00	00	00	63	7B	9A	5F	9E	5C	C6	00	00	00	20	em...c{š_ž\Æ...
00000001C0	21	00	27	DD	1E	3F	00	08	00	00	00	A0	0F	00	80	DD	!.'Ý.?..... ..€Ý
00000001D0	1F	3F	27	7A	3B	7F	00	A8	0F	00	00	98	0F	00	00	7A	.?'z;..¨...˜...z
00000001E0	3C	7F	07	FE	FF	FF	00	40	1F	00	14	EA	E1	39	00	FE	<..þÿÿ.@...êá9.þ
00000001F0	FF	FF	27	FE	FF	FF	00	30	01	3A	00	F8	36	00	55	AA	ÿÿ'þÿÿ.0.:.ø6.Uª

Partitionstabelle

MBR - Beispiel

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0000000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00	3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
0000000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00	.¹..üó¤Ph..ËÛ¹..
0000000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10	½¾.€~..|.....ƒÅ.
0000000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00	âñÍ.ˆV.UÆF..ÆF..
0000000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09	´A»ªUÍ.]r..ûUªu.
0000000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74	÷Á..t.þF.f`€~..t
0000000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00	&fh....fÿv.h..h.
0000000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13	|h..h..´BŠV.‹ôÍ.
0000000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00	ŸƒÄ.žë.¸..».|ŠV.
0000000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE	Šv.ŠN.Šn.Í.fas.þ
00000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84	N.u.€~.€.„Š.²€ë„
00000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55	U2äŠV.Í.]ëž.>þ}U
00000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64	ªunÿv.è..u.ú°Ñæd
00000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75	èƒ.°ßæ`è|.°ÿædèu
00000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54	.û¸.»Í.f#Àu;f.ûT
00000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00	CPAu2.ù..r,fh.».
0000000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66	.fh....fh....fSf
0000000110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66	SfUfh....fh.|..f
0000000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD	ah...Í.Z2öê.|..Í
0000000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4	. ·.ë. ¶.ë. µ.2ä
0000000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD	...‹ð¬<.t.»..´.Í
0000000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8	.ëòôëý+Éädë.$.àø
0000000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69	$.ÃInvalid parti
0000000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72	tion table.Error
0000000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69	loading operati
0000000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E	ng system.Missin
00000001A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
00000001B0	65	6D	00	00	00	63	7B	9A	5F	9E	5C	C6	00	00	00	20	em...c{š_ž\Æ...
00000001C0	21	00	27	DD	1E	3F	00	08	00	00	00	A0	0F	00	80	DD	!.'Ý.?..... ..€Ý
00000001D0	1F	3F	27	7A	3B	7F	00	A8	0F	00	00	98	0F	00	00	7A	.?'z;..¨...˜...z
00000001E0	3C	7F	07	FE	FF	FF	00	40	1F	00	14	EA	E1	39	00	FF	<..þÿÿ.@...êá9.ÿ
00000001F0	FF	FF	27	FE	FF	FF	00	30	01	3A	00	F8	36	00	55	AA	ÿÿ'þÿÿ.0.:.ø6.Uª

Magic Number

Exkurs: Master Boot Routine I

Windows MBR

```
seg000:0000     xor     ax, ax          ; Accumlator register auf 0
seg000:0002     mov     ss, ax          ; stack segment pointer auf 0
seg000:0004     mov     sp, 7C00h       ; Stack pointer auf 7C00h
seg000:0007     mov     es, ax          ; extra segment register auf 0
seg000:0009     mov     ds, ax          ; Data segment register auf 0
seg000:000B     mov     si, 7C00h       ; source index auf 0x7c
seg000:000E     mov     di, 600h        ; destination index auf 600h
seg000:0011     mov     cx, 200h        ; Counter register auf 200h (=512)
seg000:0014     cld                     ; “clear direction flag”
seg000:0015     rep movsb               ; von si nach di kopieren
seg000:0017     push    ax              ; wird neuer code segment pointer
seg000:0018     push    61Ch            ; wird neuer Instruction pointer
seg000:001B     retf                    ; letzen 2 Werte als neuen Code
seg000:001C     sti                     ; aktivieren von Interrupts
seg000:001D     mov     cx, 4           ; counter register auf 4
seg000:0020     mov     bp, 7BEh        ; Addresse der Partitionstabelle
```

Das BIOS kopiert den MBR an dieser Stelle

Kopieren von 512 Byte (MBR)

Maximale Anzahl Partitionen

440 + 4 + 2 = 1BEh
600h + 1BEh = 7BEh

Exkurs: Master Boot Routine II


```
seg000:0023 loc_23:
seg000:0023          cmp     byte ptr [bp+0], 0
seg000:0027          jl      short loc_34
seg000:0029          jnz     loc_13B
seg000:002D          add     bp, 10h
seg000:0030          loop    loc_23
```

Anfang der Partitionstabelle prüfen

Wenn erstes Byte der Partitionstabelle nicht 0 (z.B. 0x80) dann Bootentry gefunden => Sprung!

1 Partitionseintrag = 16 Byte (10h)
Ein Eintrag weiter gehen und wieder neu prüfen

Weder 0x80 noch 0x00? FEHLER

```
seg000:013B loc_13B:
seg000:013B                 mov     al, ds:7B5h
```
=> “Error loading operating system”

Exkurs: Master Boot Routine III

Windows MBR

```
seg000:0034 loc_34:
seg000:0034             mov     [bp+0], dl
seg000:0037             push    bp
seg000:0038             mov     byte ptr [bp+11h], 5
seg000:003C             mov     byte ptr [bp+10h], 0
seg000:0040 loc_40
seg000:0040             mov     ah, 41h ; 'A'
seg000:0042             mov     bx, 55AAh
seg000:0045             int     13h             ; DISK - Check for INT 13h Extensions
seg000:0045                                     ; BX = 55AAh, DL = drive number
seg000:0045                                     ; Return: CF set if not supported
seg000:0045                                     ; AH = extensions version
seg000:0045                                     ; BX = AA55h
seg000:0045                                     ; CX = Interface support bit map
seg000:0047             pop     bp
seg000:0048             jb      short loc_59
seg000:004A             cmp     bx, 0AA55h
seg000:004E             jnz     short loc_59
seg000:0050             test    cx, 1
seg000:0054             jz      short loc_59
seg000:0056             inc     byte ptr [bp+10h]
seg000:0059
```

Speichern der Festplattennummer (die erste ist 0x80)

werden später als Überprüfung genutzt

Register AH auf 41 == Funktion „Check Extenstion Present“ => 13h Interrupt um auf Festplatte zuzugreifen

INT 13h supperted => CF = 0 => below 1 => LADEN

Weitere Tests, falls 13h nicht supported

Exkurs: Master Boot Routine IV

Windows MBR

```
seg000:0059 loc_59:
seg000:0059             pushad
seg000:005B             cmp     byte ptr [bp+10h], 0
seg000:005F             jz      short loc_87
seg000:0061             push    large 0
seg000:0067             push    large dword ptr [bp+8]
seg000:006B             push    0
seg000:006E             push    7C00h
seg000:0071             push    1
seg000:0074             push    10h
seg000:0077             mov     ah, 42h ; 'B'
seg000:0079             mov     dl, [bp+0]
seg000:007C             mov     si, sp
seg000:007E             int     13h
seg000:0080             lahf
seg000:0081             add     sp, 10h
seg000:0084             sahf
seg000:0085             jmp     short loc_9B
```

; Start Sector der Partition steht im Partitionentry…
; … ab Byte 8 (2x 16 Bit push = 32Bit Adresse)
; offset zum aktuellen Segement in das gelesen wird
; Anzahl der zu lesenden Sectoren
; Reserved
; Größe des Disk Address Package (Summe Stack)

Hat sich unser Test geändert? => Nein => 13h Interrupt fehlgeschlagen (Try Legacy Code)

13h Interrupt (Funktion „42h“) Liest die angegebene Adresse von der Festplatte

Register sichern und zum nächsten Eintrag

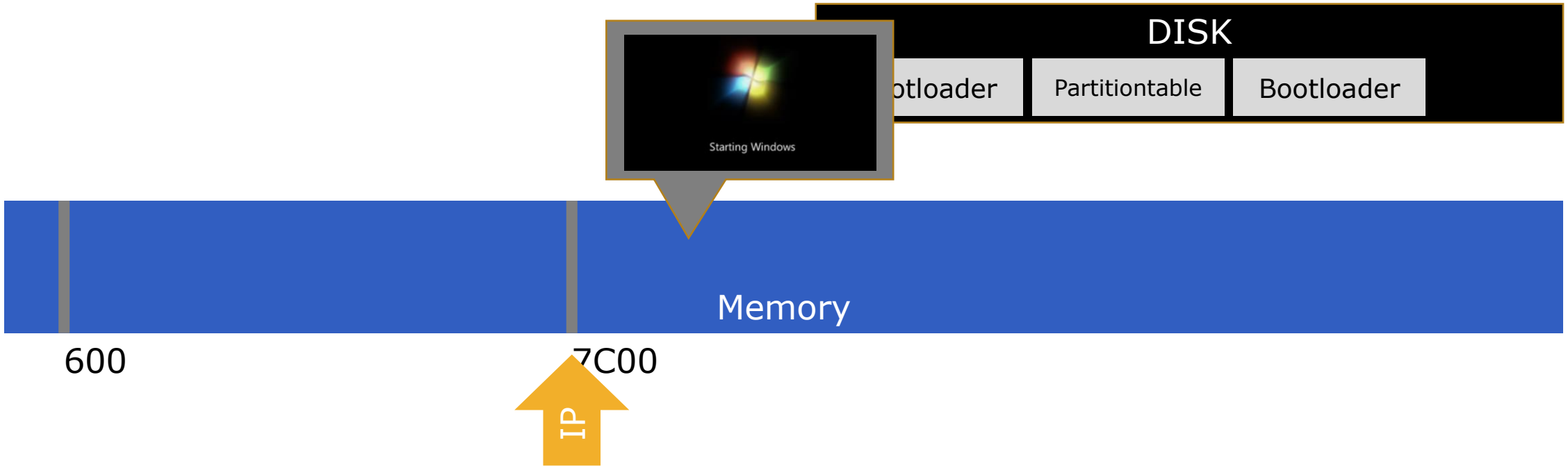
Diverse Tests (z.B. hat VBR 0xAA55h? Keyboard aktiv? TPM aktiv?) dann laden des OS

Exkurs: Master Boot Routine V


```
seg000:0127 loc_127:
seg000:0127             pop     dx
seg000:0128             xor     dh, dh
seg000:012A             jmp     far ptr 0:7C00h
```

Data Register leeren

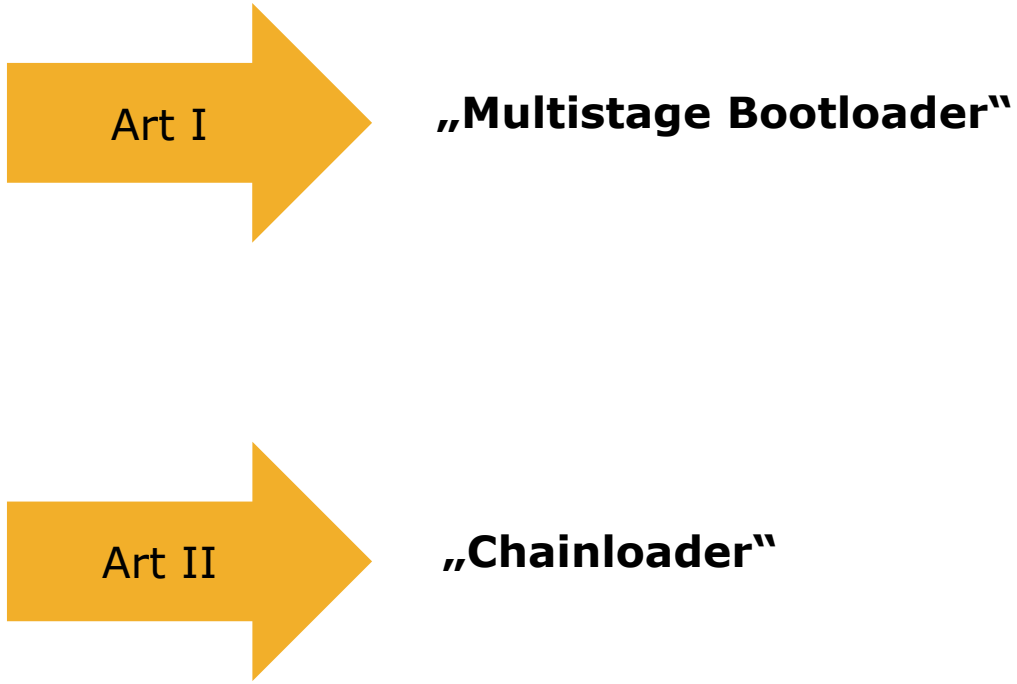
Zu OS Bootloader an 7C00h spring



Bootloader I

Grundlagen

- wird durch BIOS geladen
- liegt auf einem Bootfähigen Medium
- sorgt für das Laden des Betriebssystems
- ursprünglich liegt Bootloader nur im MBR
  - wird heute zwischen MBR und Partitionen / Dateien aufgeteilt



Bootloader II

„Multistage Bootloader“

(Mehrstufige Bootloader)

- Bootloader wird in Stufen geteilt
- Jede Stufe wird von der vorherigen geladen
- Pro Stufe können div. Funktionen ergänzt / genutzt werden (z.B. ab gewisser Stufe Bootloader via Dateinamen finden)
- Oft genutzt, wenn MBR nicht genug Platz hat

„Chainloader“

- Mehrere Bootloader laden sich nacheinander
- Vorgang als „Chainloading“ bezeichnet
- Bootloader können auf unterschiedlichen Partitionen / Medien liegen
- Beispiel aus der Praxis – GRUB
  - 1. Bootloader = Zeigt Bootmenü
  - 2. Bootloader = entsprechend ausgewählter Booteintrag
- Beispiel aus der Praxis – OS Installations CDs
  - CD startet und installiert OS -> Reboot
  - CD startet wieder, findet aber OS -> Boot from Disk

Sicherheit

„Chainloader“

- Mehrere Bootloader laden sich nacheinander

Bootloader müssen an bestimmten Stellen stehen

Virus kann diese Stelle überschreiben!

Boot Virus

- die älteste Form der Computerviren
- Nisten sich im MBR / Bootsektor ein
- Aktuell kaum noch eine Bedrohung, da OS & BIOS genug Schutz bieten

Bootkit

- Mischung aus Rootkit und Boot Virus
- Idee: Wenn man die Hardware unter Kontrolle hat, kann man auch die Software kontrollieren
- Versucht die Sicherheitsmechanismen des OS bereits beim Booten zu deaktivieren

Virtual Machine Based Rootkit

- Idee: Beim Starten, wird OS in eine VM geladen, sodass OS VMBR nicht erkennen kann -> VMBR kann aber Aktivitäten des OS aufzeichnen!
- Aktuell nur Versuche: SubVirt (Microsoft) und BluePill (in Anlehnung an Matrix)

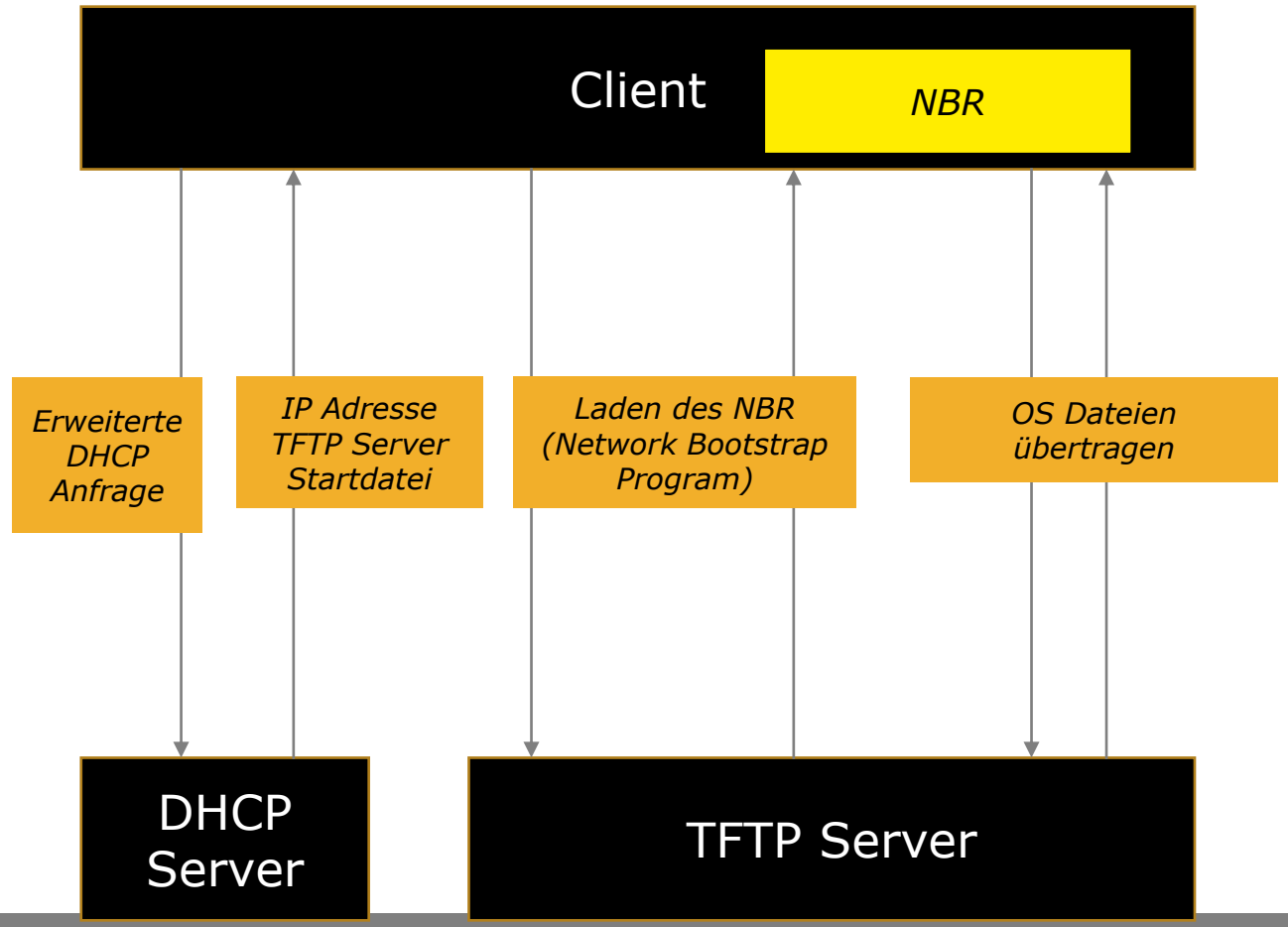
Network Boot - PXE

Preboot eXecution Environment

- Ermöglicht booten aus dem Netzwerk
- Kein lokales Medium notwendig
- Netzwerkkonfiguration via DHCP
- Voraussetzung: Netzwerkkarte ist PXE fähig

Anwendungsfälle

- Thin Clients
- OS Installation
- Wartung / Recovery



Wake on Lan I

Grundlagen

- 1995 von AMD & HP veröffentlichter Standard
- kann ausgeschaltete Computer über Netzwerkkarte starten
- Motherboard, BIOS & Netzwerkkarten müssen WOL unterstützen

Funktionsweise

- Netzwerkkarte wartet auf „Magic Packet“
  - 6x 0xFF
  - 16x MAC Adresse
  - Broadcast Adresse
- Wenn Paket empfangen, wird Start initiiert

Alternativen

- Netzwerkkarte wartet „Link State Change“
  - Reaktion auf Änderung der Netzwerkverbindung
- Pattern Match Methode
  - Reaktion auf bestimmt gerichtete Pakete (z.B. ICMP – Ping)
  - Kann zu häufigem (ungewolltem) Starten führen

Wake on Lan II