

Computersystemen 2

Theorie

2. Booten

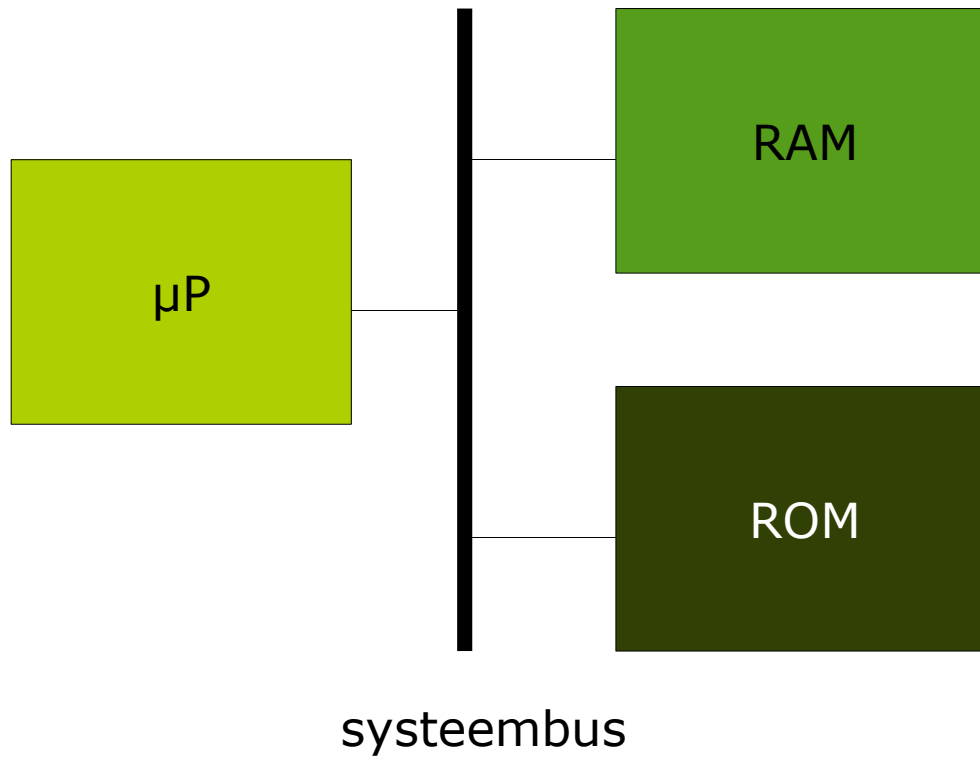
Inhoud

- ROM firmware
- Booten
- MBR en boot-loader
- UEFI
- Mogelijke vragen

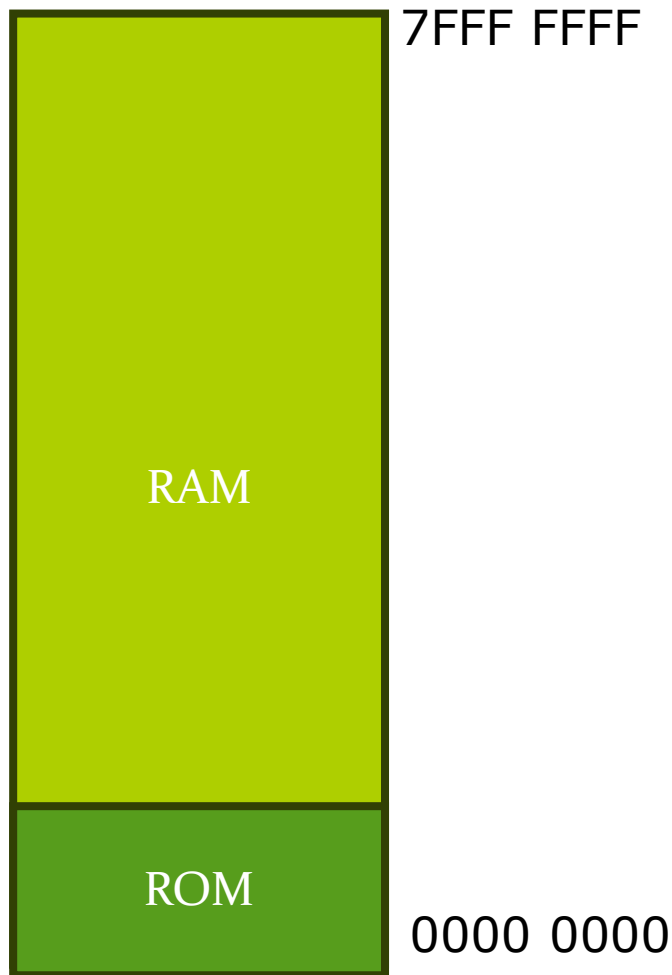
Booting

- Ik zet deze computer aan
- Wat gebeurt er?
- Wat is het probleem?
- Oplossing?

Booting



Booting



- Wat zit er in de ROM?
 - PowerOn Self Test (POST)
 - Hardware Abstraction Layer (HAL)
 - Shell (interface voor gebruiker)
 - Code om andere code te laden van extern medium (bv. harde schijf)

Inhoud

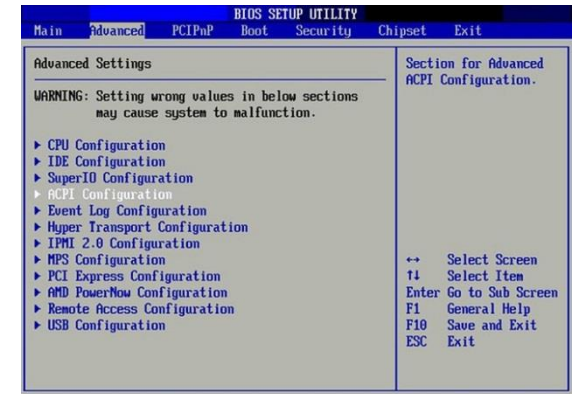
- ROM firmware
- Booten
- MBR en boot-loader
- UEFI
- Mogelijke vragen

Booting

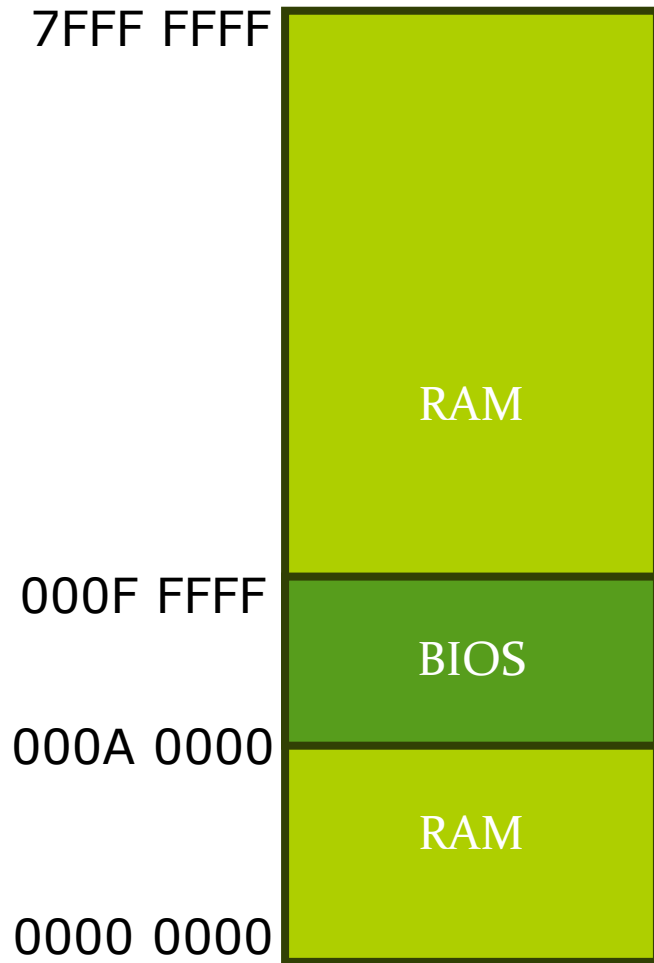
1. ROM firmware
2. Boot loader: als laatste stap laadt de firmware de boot loader van de startup disk naar RAM en voert deze uit
3. Kernel: boot loader start kernel na het laden van disk naar RAM
4. Volledig OS: processen opstarten, filesystemen mounten, netwerk configureren,...

Booting - voorbeeld x86

- PC architectuur:
 - Firmware = **BIOS**
 - BIOS: Basic input/output system
 - gemapt tussen A0000 en FFFFF
 - bevat drivers voor toetsenbord, scherm, HDD
- BIOS voert volgende stappen uit
 - POST
 - HAL
 - laad boot-loader van HDD in RAM op 7C00 en voert uit



Booting - voorbeeld x86



Midden in het RAM
geheugen gemapt??

BIOS HAL

- BIOS levert functies via "API"
- geïmplementeerd met "software interrupts"
 - instructie INT
 - http://en.wikipedia.org/wiki/BIOS_interrupt_call
 - parameters worden via registers doorgegeven! -> wat is het gevolg hiervan???
- voorbeelden
 - zet karakter op scherm (INT 0x10)
 - lees karakter van toetsenbord (INT 0x16)
 - lees **sector**¹ van HDD (INT 0x13) – BIOS kan enkel sectoren lezen, geen files!

¹ een HDD is opgedeeld in sectoren van 512 byte, zie later

Inhoud

- ROM firmware
- Booten
- MBR en boot-loader
- UEFI
- Mogelijke vragen

Booting – voorbeeld x86

- laatste stap van BIOS:
 - lees eerste sector van HDD of memory stick
 - zet deze op plaats 7C00
 - spring naar 7C00
- eerste sector = **Master Boot Record (MBR)**
 - Eerste 512 bytes van HDD
 - boot-loader (bv. grub): code die het OS laadt, met eventueel menu om verschillende OS-en te kunnen starten
 - gegevens over HDD (grootte, partities, ...)
 - bootsignatuur (magic number)

Booting - laden van OS

- Computer start op via BIOS
- Eerste sector van HDD wordt geladen, maar die is heel klein (512 bytes)
- Laad rest van boot-loader uit de volgende sectoren
- Boot-loader bevat code om een filesysteem te lezen en kan dan de kernel van het OS van de HDD laden

MBR Lab

- Bekijk 1^{ste} sector van je harde schijf of memory stick (PC)
- Linux:
`sudo dd if=/dev/sda bs=512 count=1 | hexdump -v`
- Windows HexEditor:
<http://mh-nexus.de/en/hxd/>
- Op welke adressen bevinden zich:
 - Code boot-loader
 - Partitietabel
 - bootsignatuur



Inhoud

- ROM firmware
- Booten
- MBR en boot-loader
- UEFI
- Mogelijke vragen

Booting - UEFI

- BIOS is verouderd, **UEFI** is nieuwe standaard
 - (U)EFI: (Unified) Extensible Firmware Interface
 - Heeft BIOS compatible mode

	BIOS	UEFI
Laad code van	MBR = 1 ^{ste} sector HDD	
Partitietabel	in MBR	
Grootte partitietabel	4 partities die elk 16 byte innemen in MBR (14 part. met extended/logische)	
Max grootte filesysteem	2TiB	
Kan secure boot	Nee	
Boot-loader nodig	Ja altijd	

Booting - UEFI

- BIOS is verouderd, **UEFI** is nieuwe standaard
 - (U)EFI: (Unified) Extensible Firmware Interface
 - Heeft BIOS compatible mode

	BIOS	UEFI
Laad code van	MBR = 1 ^{ste} sector HDD	EFI-file op EFI systeem partitie (bv. \EFI\BOOT\BOOTx64.EFI)
Partitietabel	in MBR	in GPT (GUID ¹ partitie tabel)
Grootte partitietabel	4 partities die elk 16 byte innemen in MBR (14 part. met extended/logische)	128 partities die elk 128 byte innemen in GPT
Max grootte filesystem	2TiB	8ZiB
Kan secure boot	Nee	Ja
Boot-loader nodig	Ja altijd	Kan rechtstreeks kernel opstarten

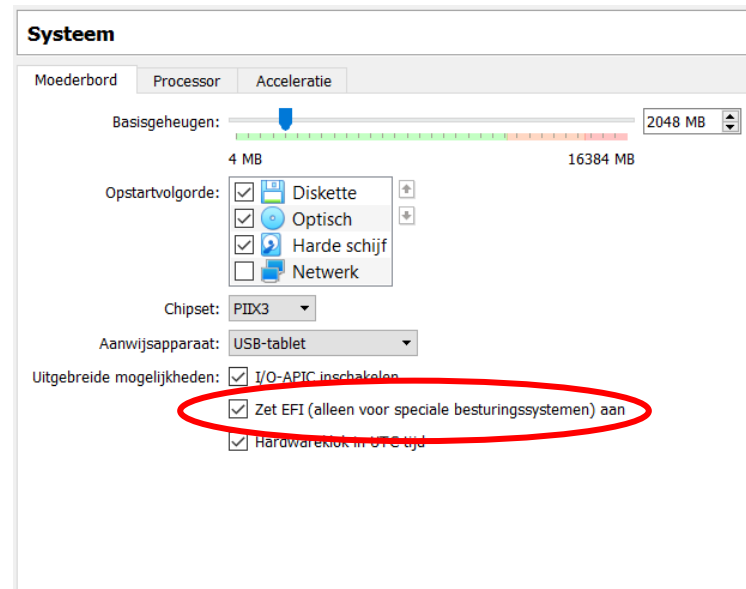
¹ GUID: Globally unique identifier

Booting lab

1. Zoek uit of jou laptop BIOS of UEFI gebruikt
2. Installeer Ubuntu in VirtualBox met UEFI
3. Ga naar de UEFI Interactive shell (ev. via de UEFI Firmware settings) door het booten te onderbreken met de Esc toets

```
CS2 [Draaiend] - Oracle VM VirtualBox
Bestand Machine Weergeven Invoer Apparaten Hulp

UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s):HD1a65535a1::BLK2:
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x0,0xFFFF,0x0)/HD(1,GPT,05D69016-3AB4-4775-80
  BLK0: Alias(s):
        PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)
  BLK1: Alias(s):
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x0,0xFFFF,0x0)
  BLK4: Alias(s):
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x1,0xFFFF,0x0)
  BLK6: Alias(s):
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x2,0xFFFF,0x0)
  BLK3: Alias(s):
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x0,0xFFFF,0x0)/HD(2,GPT,E0CD2B40-C87E-4FB9-B4
  BLK5: Alias(s):
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x1,0xFFFF,0x0)/HD(1,GPT,B000024F-A22B-054A-95
  BLK7: Alias(s):
        PciRoot(0x0)/Pci(0xD,0x0)/Sata(0x2,0xFFFF,0x0)/HD(1,GPT,E00F6CB8-ACCA-5749-B4
Press ESC in 2 seconds to skip startup.nsh or any other key to continue.
Shell> fs0:
FS0:> ls
Directory of: FS0:\
09/14/2020 11:18 <DIR>          4,096 EFI
0 File(s)                  0 bytes
1 Dir(s)
FS0:> _
```



Inhoud

- ROM firmware
- Booten
- MBR en boot-loader
- UEFI
- Mogelijke vragen

Voorbeelden van examenvragen

- Wat zijn de taken van een besturingssysteem?
- Wat is een Von Neumann architectuur?
- Wat is het verschil met een Harvard architectuur?
- Wat is een bus?
- Wat zijn I/O devices?
- Wat is POST?
- Wat is een HAL?
- Welke stappen worden uitgevoerd bij het opstarten?
- Hoe ziet de MBR eruit?
- Wat zijn de verschillen tussen BIOS en UEFI?
- Is een boot-loader nodig?
- Wat is het nut van de boot signature?
- Hoe ziet de partitietabel eruit?
- Hoe kan je zien of BIOS of UEFI hebt?