

Operating Systems

D. Leeuw

3 september 2025

v.0.7.1



© 2023 Dennis Leeuw

Dit werk is uitgegeven onder de Creative Commons BY-NC-SA Licentie en laat anderen toe het werk te kopiëren, distribueren, vertonen, op te voeren, en om afgeleid materiaal te maken, zolang de auteurs en uitgever worden vermeld als maker van het werk, het werk niet commercieel gebruikt wordt en afgeleide werken onder identieke voorwaarden worden verspreid.

Over dit Document

0.1 Voorkennis

Om dit document goed te kunnen begrijpen is de volgende kennis van belang:

- Kennis van de werking van het CPU, het geheugen de onderlinge interactie
- Kennis van de werking van de harddisk en/of SSD
- Basis kennis van software ontwikkeling maakt het makkelijker om te begrijpen, maar is niet essentieel.

0.2 Leerdoelen

Na het bestuderen van dit document heeft de lezer kennis van:

- wat een operating system is
- wat een kernel doet
- waar drivers voor zijn
- het verschil tussen programma's, applicaties, commando's en services
- de verschillende gebruikersinterfaces

Inhoudsopgave

Over dit Document	i
0.1 Voorkennis	i
0.2 Leerdoelen	i
1 Inleiding	1
2 Computer hardware	3
3 Bootstrapping / Booting	5
3.1 Master Boot Record	5
3.2 Boot Manager	5
4 De kernel	7
4.1 Drivers	7
5 User Interface	9
5.1 CLI	9
5.2 GUI	10
5.3 Desktop	10
6 Processen	11
7 Filesystems	13
Index	15

Hoofdstuk 1

Inleiding

Een operating system (OS) of in het Nederlands besturingssysteem is een complex geheel van verschillende stukjes software. Er is niet zoiets als **het** operating system. We spreken wel van Mac OS X, Windows of Linux, maar eigenlijk zijn dit samenraapsels van verschillende stukken software. De taak van het besturingssysteem is het aansturen van de hardware in opdracht van programma's of gebruikers.

Een gebruiker is een persoon, mens, die via een input-device de computer een opdracht geeft om iets uit te voeren. Een programma is een stuk software dat geschreven is door een programmeur en dat bepaalde opdrachten uitvoert in een bepaalde volgorde. Er zijn een aantal specifieke vormen van programma's:

Applicatie Een programma met een grafische interface, die door een gebruiker gebruikt kan worden. Een applicatie werkt op een desktop omgeving.

Commando Een programma zonder een grafische interface, die door een gebruiker gebruikt kan worden. Een commando werkt op de command line interface (CLI).

Service Een service is een programma dat op de achtergrond draait en daar bepaalde functies uitvoert zonder dat de gebruiker zich ermee hoeft te bemoeien.

Hoofdstuk 2

Computer hardware

Een computer is een apparaat dat data verwerkt. Om data te kunnen verwerken moeten we de computer opdrachten geven en daarvoor gebruiken we tegenwoordig programma's. Een computer kunnen we simpel beschouwen als opgebouwd uit 2 samenhangende onderdelen. We hebben de CPU (Central Processing Unit of processor) en het RAM (Random Access Memory , geheugen). In het geheugen zitten de programma's en de data. De CPU is degene die op basis van opdrachten van een programma data bewerkt uit het geheugen. Het RAM is vluchtig, dat betekent dat als we de computer uitzetten het geheugen leeg wordt. Alle programma's en alle data zijn dan verdwenen uit het geheugen. Willen we onze data (documenten) bewaren dan zullen we ze moeten opslaan op een apparaat dat niet vluchtig is. We slaan ze op op een opslagapparaat (storage device) zoals een harddisk, SSD of een USB-stick. Ook onze programma's zijn opgeslagen op een storage device en moeten om te kunnen worden gebruikt eerst vanaf het storage device in het geheugen geladen worden voordat we ze hun werk kunnen laten doen.

Hoofdstuk 3

Bootstrapping / Booting

Het is het BIOS/UEFI dat zorg draagt voor de hardware initialisatie en het is uiteindelijk het operating system (besturingssysteem) dat zorgt voor de interface naar de gebruiker en de applicaties. Het proces vanaf het moment dat het BIOS/UEFI klaar is tot het moment dat het operating systeem geladen is heet booting of bootstrapping. Dit bootstrapping proces bestaat uit verschillende stappen (fases):

1. Het laden van het master-boot record
2. Het starten van de boot loader/boot manager
3. Het laden en uitvoeren van de kernel

3.1 Master Boot Record

De laatste taak van het BIOS/UEFI is het laden van de Master Boot Record, ofwel de MBR. Het MBR is de eerste sector op de bootdisk (512 bytes). De bootdisk kan een USB-stick zijn, een SD-kaart, een SSD of harddisk. Het boot device wordt opgegeven in het BIOS en kan daar door de gebruiker gewijzigd worden.

In het master boot record staat wat de layout is van een disk (partitie-tabel) en welke partitie de kernel van het OS bevat. Het laden van de MBR en daaruit de juiste informatie halen is fase 1 van het boot proces.

3.2 Boot Manager

De boot manager is de 2de fase van het boot proces. De meest simpele oplossing is dat de boot manager de kernel van de disk laadt en deze uitvoert.

Meestal zit het geheel iets complexer in elkaar. De boot manager kan bijvoorbeeld verschillende opties geven om speciale kernels te laden. Bijvoorbeeld de keuze tussen het opstarten van Linux of Windows. Ook kan de keuze geboden worden om het standaard OS te laden of een rescue omgeving. Er zijn vele varianten mogelijk, maar uiteindelijk is het de taak van de boot manager om een kernel van disk te laden en deze te executeren. Daarna is het aan het operating system.

Hoofdstuk 4

De kernel

De kernel van een besturingssysteem is het deel dat de basis vormt. De kernel zorgt voor de aansturing van memory en CPU, het handelt interrupts af en zorgt voor het doorzetten van data naar de verschillende drivers. De kernel is de regelneef van het besturingssysteem. De kernel biedt interfaces, zogenaamde application programmable interfaces (APIs) aan aan de overige software op het systeem. Het nut hiervan is er een uniforme interface is voor de applicaties en dat de complexiteit van de hardware wordt afgeschermd. Er zijn bijvoorbeeld honderden verschillende monitoren, het is makkelijk als de applicatie tegen de kernel kan zeggen beeld dit af en dat het OS zorgt dat de data op de juiste manier bij het scherm terecht komt. Het zou vervelend zijn als elke programmeur van elke applicatie elke verkrijgbare monitor zou moeten ondersteunen vanuit zijn applicatie. Daarom zorgt de kernel voor uniforme interfaces.

Een kernel kan monolitisch zijn of modulair. Een monolitische kernel heeft alle drivers die het nodig heeft in de kernel zitten en een modulaire kernel heeft de drivers als modules los op een (boot)disk en deze laadt alleen de modules die het nodig heeft in het geheugen.

Een besturingssysteem bestaat dus uit de kernel en de drivers. Daarnaast zijn er vaak ook nog services of daemons die in het geheugen draaien en die bepaalde taken uitvoeren. Deze services of daemons zijn vergelijkbaar met gewone programma's, maar deze services behoren bij het operating system zoals deze uitgeleverd wordt door een leverancier.

4.1 Drivers

Drivers zijn specifieke stukjes software die een bepaald deel van de hardware aansturen. Bekende voorbeelden zijn drivers om printers aan te sturen of

drivers om het scherm aan te sturen. Deze stukjes software zijn soms heel specifiek zoals een driver voor een HP Color Laser 150 NW of kunnen generiek zijn zoals de driver die printers aanstuurt. Verschillende drivers kunnen dus met elkaar samenwerken. Via de printerdriver kunnen specifieke zaken geregeld worden die bij de specifieke hardware horen, zoals welke papierlade van de printer er gebruikt moet worden.

De print-API neemt de job aan, de printer driver zorgt voor het generieke regelwerk en de HP Color Laser 150 NW driver zorgt ervoor dat data wordt opgemaakt voor deze specifieke printer en stuurt de printer aan.

Hoofdstuk 5

User Interface

Vaak willen we als mensen computers opdrachten geven. Om dit te kunnen doen moeten we instaat zijn om de computer te besturen, via een muis, toetsenbord of touchscreen. Naast deze input devices moeten we weten hoe we de computer een opdracht geven, door een commando in te typen of door op een bepaald icoontje te klikken. Tot slot moet de computer terug geven wat het resultaat is van de handeling, er moet output zijn naar bijvoorbeeld een scherm of een printer.

Dit alles heeft te maken met de user interface (UI). Dit is de interface naar de gebruiker, dit in tegenstelling tot de interface naar de programma's (API). De UI gebruikt de API's om tegen de kernel te praten.

5.1 CLI

De Command Line Interface (CLI) is een van de oudste gebruikersinterfaces. De CLI geeft de gebruiker de mogelijkheid om via commando's de computer opdrachten te geven. Om commando's uit te laten voeren heeft de gebruiker een interface nodig. De algemene naam voor deze interface is de command interpreter. De command interpreter is een software laagje tussen de kernel en de gebruiker. De gebruiker typt een commando in, de interpreter voert het uit en de kernel zorgt dat het allemaal goed komt. Elk OS heeft zijn eigen command interpreter(s). Voor Windows hebben we `cmd` of `PowerShell` en voor Linux en Mac OS X hebben we de shell. Er zijn vele verschillende soorten shells, maar de meest gebruikte is `bash`.

De command line is dus een collectie commando's die door een command interpreter verwerkt wordt. Het vraagt van de gebruiker dat deze de commando's moet kennen en moet weten hoe deze te gebruiken. De leercurve voor het beheersen van deze interface is dan ook vaak stijl aan het begin. Je

moet in het begin heel veel leren voordat je iets met de computer kan doen.

5.2 GUI

De meeste desktop en laptop computers zijn voorzien van een operating system met een Graphical User Interface (GUI). Een GUI wordt aangestuurd met een pointing device (o.a. muis of trackpad) of een touchscreen. Een grafische interface is makkelijk aan te leren omdat het de mogelijkheid geeft om via plaatjes aan te geven wat een bepaalde functionaliteit doet. Een plaatje met een harddisk slaat informatie op. Een gebruiker hoeft dus geen commando's meer uit zijn hoofd te leren. De ultieme interface zou uit alleen plaatjes bestaan die weergeven wat de functionaliteit is. Dit is echter niet het geval. De meeste interfaces maken nog gebruik van menu's waar functies in zijn ondergebracht, het is dus voor de gebruiker nog noodzakelijk om te weten onder welk menu hij welke functie kan vinden.

5.3 Desktop

Een grafische interface met alleen een pointer heeft niet veel nut. We willen graag de mogelijkheid hebben om applicaties op te starten, een klok om de tijd te zien en een mogelijkheid om bestanden te benaderen. Deze totale omgeving wordt de desktop of het bureblad genoemd.

De desktop is wat de meeste mensen kennen als de GUI. De desktop is meer dan alleen een grafische interface. De desktop is dus een collectie applicaties netzo als de CLI een collectie commando's is.

Hoofdstuk 6

Processen

Operating System processen zijn diensten die in het geheugen draaien en die bepaalde taken van het operating system uitvoeren die niet in de kernel zitten. Deze processen heten soms services (Windows) en soms daemons (Unix, Linux, Mac OS X). Een print service kan zo'n dienst zijn. Deze neemt de te printen opdracht aan van een applicatie of een gebruiker en geeft deze door aan de driver om te zorgen dat die uitgeprint wordt op het juiste apparaat. De reden om een print service te hebben kan zijn omdat er meerdere gebruikers tegelijk een opdracht naar een printer zouden kunnen sturen. De print service (print server) zorgt er dan voor dat de opdrachten keurig na elkaar op de printer terecht komen.

Het is afhankelijk van het besturingssysteem welke processen onderdeel zijn van het operating system en welke niet. Die keuzes verschillen per leverancier.

Hoofdstuk 7

Filesystems

Data, besturingssysteem en programma's zijn opgeslagen op een storage device. Om enige ordening aan te brengen aan hoe we data op een storage device opslaan gebruiken we een bestandssysteem. Dit maakt het mogelijk om data op te slaan als bestanden en deze te ordenen in directories. Veel gebruikte synoniemen voor een directory zijn een folder of een map.

Een bestandssysteem is in het Engels een file system. Bijna elke operating system heeft zijn eigen file system en sommige operating systems ondersteunen meer dan één bestandssysteem. Windows gebruikt tegenwoordig NTFS, maar vroeger het FAT-filesysteem. Linux kent het extended filesystem (extfs), maar ook bijvoorbeeld BTRFS en Reiserfs. Apple gebruik het Apple File System (APFS).

Je kan geen Apple APFS of Linux extfs disk lezen op een Windows systeem. Dat maakt de uitwisseling van data lastig. Het FAT file system is al heel oud en er is voor bijna elk operating systeem inmiddels wel een driver die het file systeem kan lezen en schrijven. Dus als je data wilt uitwisselen tussen de verschillende systemen via bijvoorbeeld een USB-stick dan is het het handigst om te zorgen dat de USB-stick gebruik maakt van het FAT file system.

Index

api, [7](#)
applicatie, [1](#)
application programmable
 interface, [7](#)

bash, [9](#)
bestanden, [13](#)
bestandssysteem, [13](#)
besturingssysteem, [1](#)
bios, [5](#)
boot device, [5](#)
boot loader, [5](#)
boot manager, [5](#)
boot proces
 fase 1, [5](#)
 fase 2, [5](#)
 fase 3, [5](#)
booting, [5](#)
bootstrapping, [5](#)
buroblad, [10](#)

central processing unit, [3](#)
cli, [9](#)
cmd, [9](#)
command, [1](#)
command interpreter, [9](#)
command line interface, [9](#)
commando, [9](#)
computer, [3](#)
cpu, [3](#)

daemons, [7](#), [11](#)

data, [3](#)
desktop, [10](#)
directory, [13](#)
documenten, [3](#)
drivers, [7](#)

file system, [13](#)
folder, [13](#)

geheugen, [3](#)
graphical user interface, [10](#)
gui, [10](#)

harddisk, [3](#)
hardware initialisatie, [5](#)

input devices, [9](#)
input-device, [1](#)
interfaces, [7](#)

kernel, [7](#)

Linux, [1](#)

Mac OS X, [1](#)
map, [13](#)
master boot record, [5](#)
mbr, [5](#)
modulair, [7](#)
monolitisch, [7](#)
muis, [9](#), [10](#)

opdrachten, [3](#)
operating system, [1](#)
opslagapparaat, [3](#)

- OS, [1](#)
- output, [9](#)
- partitie-tabel, [5](#)
- pointing device, [10](#)
- PowerShell, [9](#)
- print server, [11](#)
- print service, [11](#)
- printer, [9](#)
- processen, [11](#)
- processor, [3](#)
- programma, [1](#), [3](#)
- programmeur, [1](#)
- RAM, [3](#)
- Random Access Memory, [3](#)
- scherm, [9](#)
- service, [1](#)
- services, [7](#), [11](#)
- shell, [9](#)
- ssd, [3](#)
- storage device, [3](#)
- toetsenbord, [9](#)
- touchscreen, [9](#), [10](#)
- trackpad, [10](#)
- uefi, [5](#)
- ui, [9](#)
- usb-stick, [3](#)
- user interface, [9](#)
- Windows, [1](#)