

# Linux introductie

D. Leeuw

10 februari 2022

v.0.9.0



© 2020-2022 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

Dit document behandelt Linux voor het middelbaar beroepsonderwijs in Nederland, maar kan breder ingezet worden, daar het gericht is op het behalen van het LPI Linux Essentials examen. De doelgroep is niveau 4 van het MBO, met enige kennis van computers.

## Versienummering

Het versienummer van elk document bestaat uit drie nummers gescheiden door een punt. Het eerste nummer is het major-versie nummer, het tweede nummer het minor-versienummer en de laatste is de nummering voor bug-fixes.

Om met de laatste te beginnen als er in het document slechts verbeteringen zijn aangebracht die te maken hebben met type-fouten, websites die niet meer beschikbaar zijn, of kleine foutjes in de opdrachten dan zal dit nummer opgehoogd worden. Als docent of student hoeft je boek niet te vervangen. Het is wel handig om de wijzigingen bij te houden.

Als er flink is geschreven aan het document dan zal het minor-nummer opgehoogd worden, dit betekent dat er bijvoorbeeld plaatjes zijn vervangen of geplaatst/weggehaald, maar ook dat paragrafen zijn herschreven, verwijderd of toegevoegd, zonder dat de daadwerkelijk context is veranderd. Een nieuw cohort wordt aangeraden om met deze nieuwe versie te beginnen, bestaande cohorten kunnen doorwerken met het boek dat ze al hebben.

Als het major-nummer wijzigt dan betekent dat dat de inhoud van het boek substantieel is gewijzigd om bijvoorbeeld te voldoen aan een nieuw kwalificatiedossier voor het onderwijs of een nieuwe versie van Linux Essentials van de LPI. Een nieuw major-nummer betekent bijna altijd voor het onderwijs dat in het nieuwe schooljaar men met deze nieuwe versie aan de slag zou moeten gaan. Voorgaande versies van het document zullen nog tot het einde een schooljaar onderhouden worden, maar daarna niet meer.

## Document ontwikkeling

Het doel is door middel van open documentatie een document aan te bieden aan zowel studenten als docenten, zonder dat hier hoge kosten aan verbonden zijn en met de gedachte dat we samen meer weten dan alleen. Door samen te werken kunnen we meer bereiken.

Bijdragen aan dit document worden dan ook met alle liefde ontvangen. Let u er wel op dat materiaal dat u bijdraagt onder de CC BY-NC-SA licentie vrijgegeven mag worden, dus alleen origineel materiaal of materiaal dat al vrijgegeven is onder deze licentie.

De eerste versie is geschreven voor het ROC Horizon College.

Versienummer	Auteurs	Verspreiding	Wijzigingen
0.1.0	Dennis Leeuw	Wim Onrust	Initieel document
0.2.0	Dennis Leeuw	HEITO18IB-A	Toegevoegd: versie-nummering, de shell, begin van werken met bestanden

Tabel 1: Document wijzigingen

# Inhoudsopgave

<b>Over dit Document</b>	<b>i</b>
<b>1 Inleiding</b>	<b>1</b>
<b>2 Wat is Linux?</b>	<b>3</b>
<b>3 Waarom Open Source?</b>	<b>7</b>
3.1 Gebruiksrechten en licenties . . . . .	9
3.1.1 MIT . . . . .	10
3.1.2 BSD . . . . .	10
3.1.3 GPL . . . . .	11
3.2 Creative Commons . . . . .	11
3.3 Open standaarden . . . . .	12
3.4 Open data . . . . .	12
3.5 Open Source Business Model . . . . .	13
<b>4 Distributies</b>	<b>15</b>
4.1 Debian . . . . .	15
4.2 Red Hat . . . . .	16
4.3 OpenSuSE . . . . .	16
<b>5 Installatie van CentOS op Virtual Box</b>	<b>17</b>
5.1 Download CentOS boot image . . . . .	17
5.2 Aanmaken van een NAT netwerk . . . . .	17
5.3 Aanmaken van een Virtual Machine . . . . .	19
5.4 CentOS installatie . . . . .	20
<b>Index</b>	<b>27</b>



# Hoofdstuk 1

## Inleiding

Deze Linux cursus beoogt aan te sluiten bij het Linux Essentials examen van de LPI (Linux Professional Institute) en dient als voorbereiding op het MBO ICT Systems and Devices Expert examen. De cursus bestaat uit drie delen in het eerste deel installeren we CentOS als workstation en leren we een grafische interface van het Linux systeem kennen. In het tweede deel werken we op de command line (CLI) en blijven we de CentOS installatie gebruiken. In deel 3 installeren we Debian en zullen we meer leren over Linux als server. De keuze om CentOS als workstation te installeren en Debian als server is volledig willekeurig. Het doel is dat de studenten kennis maken met de rpm en dpkg package managers en leren dat het ene Linux systeem het andere niet is.

Alle Linux systemen zullen geïnstalleerd worden als virtuele machines op Virtual Box (<https://www.virtualbox.org/>). Door gebruik te maken van virtuele machines zijn we niet afhankelijk van de onderliggende hardware. De keuze voor VirtualBox heeft te maken met het feit dat dit product gratis te gebruiken is en beschikbaar is voor zowel Windows, Mac OS X als Linux.

Voor de CentOS machine is 15G vrije schijfruimte nodig en voor het Debian systeem 8G, wat een totaal aan 23G vrije schijfruimte vereist. Voor elke machine hebben we 2G RAM nodig, dus een totaal van 4G RAM moet vrij beschikbaar zijn.





## Hoofdstuk 2

### Wat is Linux?

Om de vraag te kunnen beantwoorden wat Linux is moeten we een stukje terug in de tijd en wel naar het eind van de jaren '60 uit de vorige eeuw. Een groepje programmeurs werkend aan een project bij Bell Labs van AT&T genaamd Multics zaten op een dood spoor of beter Bell Labs vond het een dood spoor en cancelde het project, dus schreef een van die programmeurs in ongeveer een maandtijd in 1969 een simpeler alternatief. De ontwikkelaar was Ken Thompson die samen met Dennis Ritchie een team leidde.

Een ander lid van het team Brian Kernighan schijnt met de naam Unics gekomen te zijn als woordgrap op Multics. Hoe de naam ooit Unix is geworden is niet bekend.

De gedachte achter Unix is ooit in 1979 mooi beschreven door Dennis Ritchie: What we wanted to preserve was not just a good environment in which to do programming, but a system around which a fellowship could form. We knew from experience that the essence of communal computing, as supplied by remote-access, time-shared machines, is not just to type programs into a terminal instead of a keypunch, but to encourage close communication.

Het ging dus om een systeem waarop samengewerkt kon worden en dan vooral geprogrammeerd.

Uit de wens vanuit de organisatie om tekst te kunnen verwerken werd het systeem uitgebreid met tekstverwerkingsfuncties en een eerste tekst editor en tekst formatter. Hieruit blijkt meteen de filosofie achter Unix. Maak kleine tools die één ding goed doen en gebruik ze gezamenlijk om complexe dingen te doen. Er is dus een editor en een formatter. De eerste tekst formatter heette roff en deze werd als snel opgevolgd door troff, die je nog steeds terug vindt op de systemen. De unix manual-pages, waarover later meer, worden opgemaakt met behulp troff.

De eerste versies van Unix waren geschreven in assembly, in 1974 kwam versie 4 uit die volledig herschreven was in de taal C. Door het gebruik van

C werd het mogelijk Unix ook op andere systemen te draaien dan het PDP systeem waar het voor geschreven was. Unix kon de wereld veroveren.

Bell Labs mocht door juridische afspraken geen commerciële zaken ondernemen buiten de telefonie, daardoor werd de vraag naar Unix beantwoord door alleen geld te rekenen voor de media en het verzenden van het systeem, en niet voor het product zelf.

Omdat het systeem goedkoop verkrijgbaar was was het een ideaal systeem voor universiteiten. Door studenten en andere gebruikers werden er in de loop van de tijd meer en meer programma's geschreven en ook anderen gingen collecties samenstellen die ze ook weer distribueerden. Deze collecties van programma's werden dan ook distributies genoemd. Een van de meest bekende van deze distributies is die van University of California die ontwikkeld werd door de Berkeley Computer Systems Research Group, de Berkeley Software Distribution of afgekort BSD.

Een klein bedrijfje in de US was een van de eerste bedrijven die Unix naar de microcomputer, ofwel de PC, bracht. Hun Unix versie heette Xenix en het bedrijf stond bekend als Microsoft. Dit naast de grote bedrijven als IBM (AIX) en HP (HPUX) die hun eigen unix versie hadden voor de grote machines.

De vele verschillende systemen die ontstonden waren voor de eindgebruikers een ramp. Wat op het ene systeem aanwezig was was op een andere niet aanwezig en omgekeerd. Een Unix-applicatie op de ene distributie werkte werkte niet zomaar ook op een andere. Gebruikers wilden standaardisatie. Software moest uitwisselbaar zijn. De standaard die ontstond in 1988 is de POSIX standaard van de IEEE.

In 1983 bracht Richard Stallman een nieuw project in de wereld genaamd het GNU Project, waarbij GNU staat voor GNU's not Unix! Dit om aan te geven dat het gebaseerd is op Unix maar geen Unix componenten bevat. GNU is een from scratch geschreven systeem dat volledig open source is. Dat wil zeggen dat van het hele systeem alle broncode beschikbaar is, hierdoor kan het gebruikt worden als studiemateriaal. Het heeft tevens het voordeel dat iedereen fouten uit de software kan halen en aan de software kan bijdragen om het beter te maken. Dit is dan ook altijd de strijd van Richard Stallman geweest en zijn daarvoor opgerichte organisatie The Free Software Foundation, software moest vrij zijn en voor iedereen toegankelijk. Om dit te bewerkstelligen stelde Stallman het copyleft in, in plaats van het copyright en maakte een licentie genaamd de GPL, GNU Public License, die ervoor moest zorgen dat de gemaakte software niet zomaar weer gesloten kon worden.

De software werd door vele verschillende distributies in die tijd gebruikt om software van Unix geheel of gedeeltelijk te vervangen. Missend onderdeel was echter jarenlang een kernel. Het stukje software dat ligt tussen de

gebruikersinterface en de hardware.

In 1991 schreef een Finse student met de naam Linus Torvalds een stukje in een news-group dat hij bezig was met zo'n kernel. Niets groots zei hij, maar wel onder de GPL-licentie van de Free Software Foundation. Velen werden aangetrokken door dit project en gingen Linus helpen zijn kernel verder uit te breiden. Die kernel ging naar de maker heten en werd Linux.

Uiteindelijk is Linux dus de abstractie laag tussen hardware en de gebruikers interface, met daarbij meestal veel van de software uit het GNU-project. Fanatiekelingen willen dan ook graag dat je spreekt van GNU/Linux, maar de hele wereld spreekt meestal gewoon over Linux als we een systeem bedoelen met een Linux kernel.

De Linux kernel kan natuurlijk gecombineerd worden met vele software pakketten en dat gebeurde dan ook. En net als bij Unix werden de systemen die ontstonden distributies genoemd. Eén van de eerste distributies was Slackware, later volgde anderen zoals SuSE, Debian, Red Hat, CentOS, Ubuntu en nog vele vele anderen.

Het meest gebruikte Unix desktop systeem is ongetwijfeld Darwin het open source besturingssysteem van Apple dat de basis vormt voor Mac OS X en op de telefoon is dat Android het systeem van Google dat een Linux kernel en andere open source tools onderwater gebruikt.



## Hoofdstuk 3

# Waarom Open Source?

Wat Linux en het GNU project bijzonder maken is het feit dat alle code open source is. Je mag er mee doen en laten wat je wil, je mag het aanpassen, je kan het inzien en je mag het gebruiken en voor het merendeel nog gratis ook. Dat is natuurlijk bijzonder in een wereld die draait om commercie. Waarom is het dan toch zo populair? Om daar een goed antwoord op te geven moeten we eerst eens kijken wat software nu eigenlijk is en hoe het werkt.

De eerste versies van Unix zoals geschreven door Ken Thompson, Dennis Ritchie en de overige leden van het team bij Bell Labs was geschreven in assembly language. Assembly language is een taal die heel dicht ligt tegen wat computers snappen en daarmee altijd systeemafhankelijk is. In de tijd dat Unix werd ontwikkeld geloofde men dat je assembly language nodig had om een besturingssysteem snel genoeg te laten zijn. Dennis Ritchie nam de taal B ontwikkeld door Ken Thompson, maakte verbeteringen en kwam met C in 1972. In 1973 kwam Unix versie 4 uit die voor een groot deel herschreven was in C en daarmee aantoonde dat een hogere programmeertaal gebruikt kon worden om besturingssystemen in te schrijven, maar belangrijker nog omdat er een hogere programmeertaal werd gebruikt was Unix opeens portable naar andere machines.

Voor het programmeren in C heb je een C-compiler, een linker en een C-Library nodig. Library is het Engelse woord voor bibliotheek een C-bibliotheek bevat een aantal standaardfuncties die je kan gebruiken in een programma, zo hoeft niet elke programmeur de printf functie te programmeren. Functies kunnen hergebruikt worden door het aanroepen van de library. Een heel simpel programma als Hello World ziet er in C zo uit:

```
#include <stdio.h>

int main() {
    printf("Hello ,_World!");
}
```

```
    return 0;  
}
```

De `printf` functie die de woorden “Hello World!” op het scherm laat zien is zo’n standaard functie uit de C-Library.

De compiler is verantwoordelijk voor het omzetten van de C-code in machinetaal, het compileren, en de linker zorgt ervoor dat de functie uit de C-library gelinkt wordt aan het programma dat je geschreven hebt. En zo kunnen we dus simpel stukjes programmatuur (functies) hergebruiken door er een library van te maken.

Er zijn twee manieren waarop de linker ervoor kan zorgen dat de `printf`-functie gelinked kan worden met je programma. Het kan statisch en dynamisch. Statisch betekent dat een kopie van de functie toegevoegd wordt aan je programma. Je programma wordt daarmee onafhankelijk van de C-library. Dynamisch linken betekent dat in je programma een verwijzing komt te staan naar de `printf`-functie in die specifieke C-library. Je C-programma wordt zo afhankelijk van de specifieke versie van de C-library die aanwezig is op je systeem. Dat is geen probleem zolang je het programma gebruikt op systemen die dezelfde C-library hebben als jij, zoals het geval is bij mensen die dezelfde distributie gebruiken als jij. Het voordeel is dat je eigen programma veel kleiner wordt en dus makkelijker te verspreiden is. Ook als er een kleine wijziging gemaakt wordt in de `printf`-functie die geen invloed heeft op de binaire syntax van de `printf`-functie dan kan je programma gelijk gebruik maken van deze verbetering doordat je alleen de C-library update. Je hoeft je programma dan niet opnieuw te bouwen, compileren.

Daar zit het verschil in API en ABI compatibiliteit. De API van een functie is de syntax van de functie, als deze verandert dan moet je je programma opnieuw compileren en als het tegenzit moet je zelfs je code aanpassen. ABI compatibiliteit is veel complexer, maar kleine wijzigingen kunnen zoals gezegd soms doorgevoerd worden zonder dat de applicatie er last van heeft.

Een programma dat alleen als binary, dus als gecompileerde versie wordt uitgeleverd kan dus alleen draaien op systemen die gelijk zijn aan het systeem waarop de code gecompileerd is, zoals op Windows en Mac OS X systemen gebeurt. Ook de applicaties op je telefoon zijn vaak afhankelijk van de juiste versie van het OS.

Als je echter de beschikking hebt over de broncode dan kan je die code ook compileren op je eigen computer en zorgen dat die ook werkt op jou systeem. Je bent dan niet meer afhankelijk van een leverancier die jou systeem moet ondersteunen. Soms moet je wel wat aanpassingen maken om het geheel goed te laten werken. De grafische interface van Windows is heel anders dan die van Mac OS X, en daar zit dan ook een heel andere library onder.

Dus als je een Windows applicatie op een Mac wil compileren zul je wel wat programmeerwerk moeten doen. Maar als ik een applicatie heb die geschreven is voor Debian kan ik hem meestal zonder enige wijziging compileren op CentOS.

En daar zit de kracht van open source door programma's, met het delen van de broncode wordt de reikwijdte van systemen waarop de software kan draaien veel groter. Maar zoals gezegd moet je soms wel wijzigingen maken om het te laten werken, dus het is bijna een eis dat je de software mag aanpassen en dat is dan ook in veel open source licenties vastgelegd.

## 3.1 Gebruiksrechten en licenties

Het auteursrecht stamt uit 1710 en was oorspronkelijk bedoeld om de drukker te beschermen. Later is dit over gegaan op de auteur. Omdat ook een programmeur een schrijver is geldt er voor broncode dezelfde rechten als op boeken. Maar wat houdt auteursrecht nu eigenlijk in.

Volgens de wet:

"Het auteursrecht is het uitsluitend recht van de maker van een werk van letterkunde, wetenschap of kunst, of van diens rechtverkrijgenden, om dit openbaar te maken en te verveelvoudigen, behoudens de beperkingen, bij de wet gesteld." (Artikel 1 Auteurswet)

Dit zegt dus iets over het openbaar maken en het kopiëren (verveelvoudigen) van het gemaakte werk. De auteur mag dit doen en niemand anders (uitsluitend recht van de maker). Als een programmeur software schrijft en dit aan iemand anders geeft, dan mag deze de software gebruiken maar niet kopiëren en verder verspreiden.

Voor de auteur zijn de regels dus in de wet vastgelegd, welke rechten een gebruiker van de software heeft dat mag de auteur zelf bepalen. Deze rechten worden vastgelegd in een gebruikerslicentie. Elke auteur kan zijn eigen licentie maken, wat veel bedrijven dan ook doen. De meest bekende is waarschijnlijk Microsoft's EULA; End-Users License Agreement. De EULA zegt in het kort dat Microsoft niet verantwoordelijk is voor de gemaakte software en eventuele fouten die het mocht bevatten, je het op 1 machine mag gebruiken, je mag geen kopieën maken en het mag maar door 1 persoon tegelijkertijd worden gebruikt.

Omdat Unix ontstaan is uit een systeem van software met elkaar delen zijn er andere licenties ontstaan. De meestvoorkomende zullen we in dit hoofdstuk behandelen.

### 3.1.1 MIT

De MIT (Massachusetts Institute of Technology) licentie geven we hier in zijn geheel weer omdat hij de basis vormt voor veel van de erop volgende licenties.

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Eigenlijk zegt deze licentie dat je alles met de software mag doen, behalve het weghalen van de copyright en de licentievoorwaarden. Ook is er een duidelijk statement dat de maker niet verantwoordelijk is voor wat de software doet. Deze laatste kom je in bijna elke licentie tegen. Ook in de EULA van Microsoft staat dit voorbehoud.

### 3.1.2 BSD

De BSD licentie, of beter de 3-clause BSD-licentie, voegt daar wat regels aan toe:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.



2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

De belangrijkste afwijking is de laatste regel, namelijk dat de naam van de schrijver van de software niet gebruikt mag worden voor promotionele doeleinden zonder schriftelijke toestemming. Dit was om te voorkomen dat een gewijzigde versie verspreid zou worden onder de naam van de oorspronkelijke auteur alsof deze de wijzigingen goed gekeurd zou hebben. Het was dus belangrijk dat er een onderscheid kwam tussen de oorspronkelijke code en de ervan afgeleide producten.

### 3.1.3 GPL

De GPL van de Free Software Foundation voegt er vele regels aan toe. De belangrijkste regel hierin is dat als je wat wijzigt in de software dan moet je die gewijzigde broncode ook weer openbaar maken. Wat bij de vorige twee licenties mogelijk is is de code te nemen, er wijzigingen in aan te brengen en dan alleen een gecompileerde (binaire) versie te leveren zonder de broncode. Dat mag met de GPL niet meer. Elke wijziging die niet alleen voor je zelf maakt, daarvan moet ook de broncode weer publiek beschikbaar zijn.

Als het goed is, althans dat is het doel van deze licentie, wordt elke verbetering zo openbaar en wordt de software beter. Je kunt een probleem dat je opgelost hebt dus niet meer alleen voor jezelf houden.

## 3.2 Creative Commons

Naast software dat speciaal is omdat het bestaat als broncode en als gecompileerde versie (executable) bestaat wordt er natuurlijk ook documentatie geschreven voor de software, worden er logo's gemaakt voor de software en websites ontworpen, soms worden er ook filmpjes gemaakt, etc. Kortom we willen vaak meer creatieve uitingen vangen in een licentie dan alleen de software. In het begin werden de software licenties ook voor deze zaken gebruikt, maar dat bleek niet altijd toereikend. De oplossing is uiteindelijk gevonden in een set van voorwaarden die bekend staan als de Creative Commons.

De Creative Commons kan je zelf samenstelling uit een aantal onderdelen en zo zelf bepalen wat er wel niet met je werk gedaan mag worden. De meest eenvoudige vorm is de CC BY. Deze zegt dat iedereen van alles met je werk mag doen, maar dat ze daarbij altijd aan naamsvermelding moeten dienen. Een beetje zoals de MIT licentie.

Zaken die je toe kan voegen zijn:

**ND** staat voor dat er Geen Afgeleide werken gemaakt mogen worden. Een logo mag dus wel gebruikt worden maar niet gewijzigd.

**NC** wat staat voor Niet Commercieel. Het product of een afgeleide ervan mag niet commercieel gebruikt worden.

**SA** staat voor dezelfde licentie. Het product of een afgeleide daarvan mag dan niet van licentie veranderen. Dit lijkt erg op de GPL licentie.

Dit boek is uitgebracht onder de CC BY NC SA. Dat wil dus zeggen dat als er een gewijzigd werk gemaakt wordt van dit document dan moeten daar de namen van de auteurs van dit document bij vermeld worden, mag het niet commercieel uitgegeven worden en moet het onder dezelfde licentie verspreid worden. Je mag dus wel geld vragen voor het feit dat je boek bijvoorbeeld gedrukt hebt.

### 3.3 Open standaarden

Een verwarring die weleens wil ontstaan is het verschil tussen open source en open standaarden en toch is daar een wezenlijk verschil.

Open standaarden beschrijven hoe bijvoorbeeld data uitgewisseld kan worden. Protocollen als SMTP, POP3, IMAP, Telnet en FTP zijn allemaal beschreven in documenten die vrij op Internet toegankelijk zijn. Iedereen, dus ook de open source wereld, kan deze standaarden implementeren en er dus voor zorgen dat verschillende systemen, open source en commercieel, met elkaar kunnen communiceren. Gesloten protocollen die door een bedrijf zijn bedacht kunnen alleen door dat bedrijf gebruikt worden, hoewel door luisteren op het netwerk er natuurlijk ook gekeken kan worden hoe het protocol werkt.

### 3.4 Open data

In navolging van open source en open standaarden kwam er ook steeds meer de vraag op bij data, kan data ook open zijn. De creative commons heeft daar

al goede aanzet voor gegeven, maar een van de belangrijkste beperkingen was lange tijd de uitwissing van data en dan voornamelijk omdat bestandsformaten van het ene systeem niet gebruikt konden worden op een ander systeem. Het uitwisselen van met name office documenten zou enorm gebaad zijn bij een open standaard. Die is dan ook ontwikkeld in de vorm van het Open Document Format.

## 3.5 Open Source Business Model

Een veel gehoord tegenargument bij het gebruik van een open source licentie bij de ontwikkeling van nieuwe software is dat er geen geld te verdienen zou zijn op deze manier. En dat is gedeeltelijk waar. Het een keer ontwikkelen van software, het daarna eindeloos kopiëren en er geld voor vragen werkt niet meer. Maar je mag nog altijd geld vragen voor de distributie, installatie hulp aanbieden, cursussen aanbieden, of adviesuren verkopen.



# Hoofdstuk 4

## Distributies

Een van de allereerste distributies was het Softlanding Linux System (SLS) door Peter MacDonald in 1992, deze Linux distributie bevatte als eerste ook een grafische interface. Het stond bekend om zijn buggy character en er ontstonden dan ook al snel opvolgers zoals Slackware van Patrick Volkerding dat in 1993 uitkwam, de oudste nog steeds bestaande distributie, en Yggdrasil. Ook Debian is een afgeleide van SLS.

Een Linux distributie is een collectie van software samen met de Linux-kernel. Veel van de software is vaak afkomstig van het GNU-project. In dit hoofdstuk geven we een overzicht wat er zich zoal in een distributie kan bevinden. Dit betekent niet dat je alle genoemde software ook daadwerkelijk in elke distributie tegenkomt. Makers van een distributie maken hun eigen keuzes welke software zij belangrijk vinden, maar met dit hoofdstuk krijg je een beeld met wat er zoals mogelijk is.

In dit boek zullen we je laten kennis maken met twee dominante Linux distributies die van Red Hat en die van Debian.

### 4.1 Debian

Ubuntu en Linux Mint zijn gebaseerd op Debian en alle drie zijn veel gebruikte distributies. Ubuntu en Linux Mint zijn vooral populair als desktop omgevingen, terwijl we Debian vaker terug vinden in een serverruimte.

Debian is een distributie die ooit is opgezet door Ian Murdoch, in 1993, en de distributie is vernoemd naar hem en zijn vrouw Debra; Deb-Ian ofwel Debian. Debian wordt ontwikkeld zoals ook open source software wordt ontwikkeld, door vrijwilligers op een volledig open en transparante manier.

## 4.2 Red Hat

Red Hat is de grootste commerciële leverancier van Linux en tegenwoordig onderdeel van IBM. Red Hat wordt in het bedrijfsleven veel gebruikt. Omdat Red Hat commercieel is moet je betalen voor gebruik. Gelukkig voor ons is er een community based version genaamd CentOS die we gratis kunnen gebruiken. CentOS is dan ook de versie die we verderop in de boek zullen installeren.

## 4.3 OpenSuSE

De SuSE Linux distributie komt oorspronkelijk uit Duitsland en is een afgeleide van Slackware. De naam was in het begin een afkorting S.u.S.E. die stond voor: Software- und SystemEntwicklung. De eerste release was in 1994 en dat maakt SuSE tot een van de oudste commerciële distributies. In 2003 is SuSE gekocht door Novell dat toen nog groot was in de server-markt. Novell is degene geweest die OpenSUSE, de niet commerciële versie op de markt gezet heeft in 2005. Daarna is SuSE nog verscheidene malen overgenomen door ander bedrijven. Op dit moment is het in handen van EQT, een Europese investeringsmaatschappij.

SuSE is vooral bekend geworden om door de YaST, Yet Another Setup Tool. De interface waarmee de Linux distributie op een eenvoudige manier beheerd en geïnstalleerd kan worden.

## Hoofdstuk 5

# Installatie van CentOS op Virtual Box

### 5.1 Download CentOS boot image

CentOS ken je download via [centos.org](http://centos.org), maar dichterbij huis is er een mirror bij de TU Delft. Ga naar <http://ftp.tudelft.nl/centos.org/>

Selecteer de meest recente 8 versie, daarna de isos map en dan de x86\_64 map.

Download het bestand met als naam:

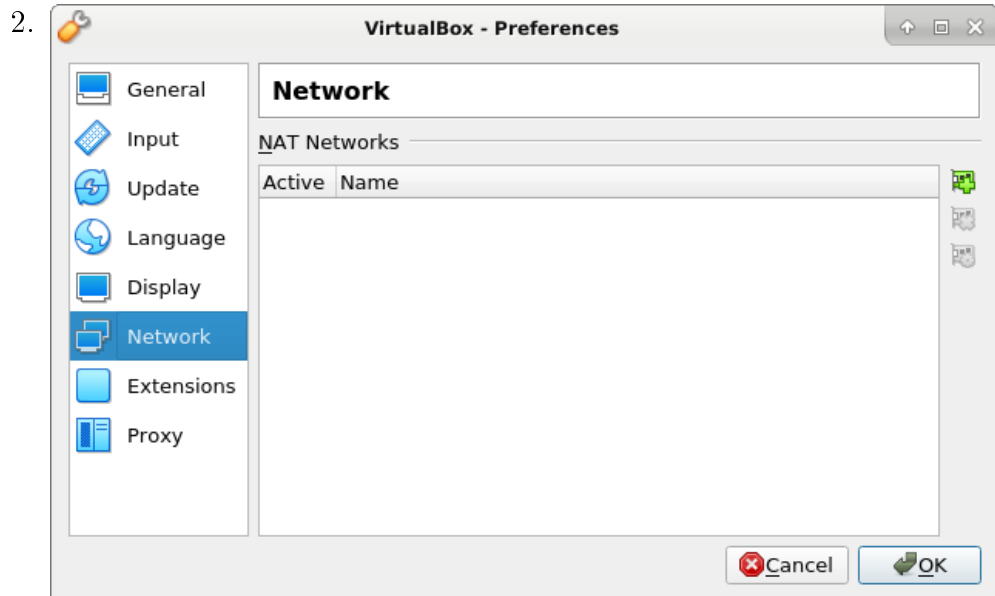
CentOS-<versie\_nummer>-x86\_64-boot.iso

Waarbij je <versie\_nummer> vervangt door het versienummer dat je wil downloaden. Terwijl dit aan het downloaden is kan je het hoofdstuk volgen over het aanmaken van een virtual machine in Virtual Box.

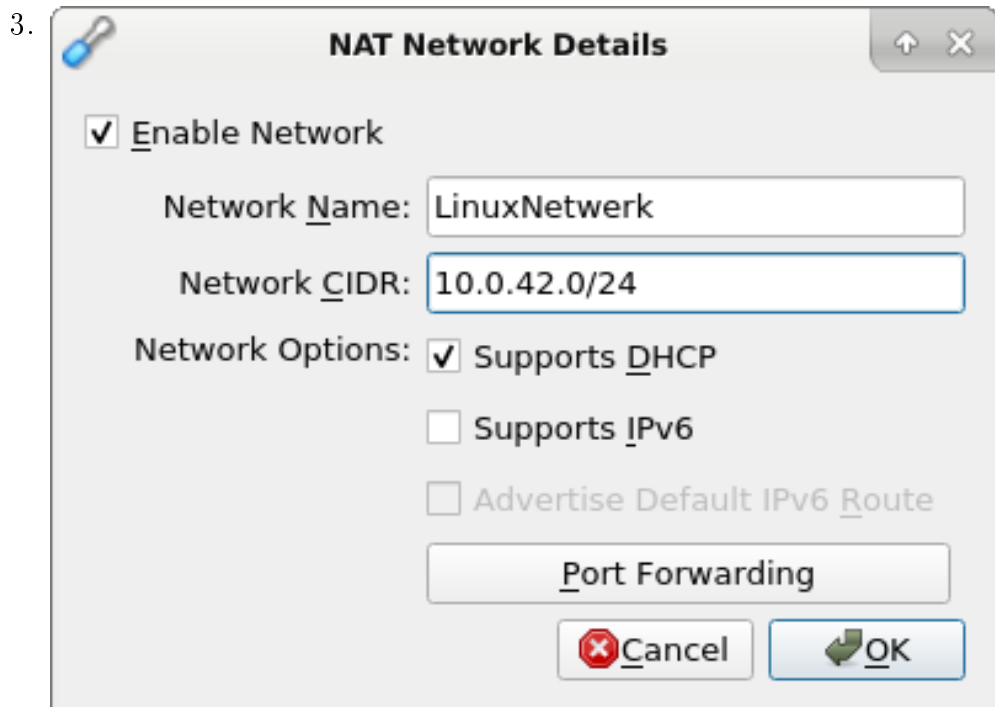
### 5.2 Aanmaken van een NAT netwerk

1. Selecteer onder het File menu de optie Preferences.

18 HOOFDSTUK 5. INSTALLATIE VAN CENTOS OP VIRTUAL BOX



Click op het groene plusje aan de rechter kant.



Geef het netwerk de naam “LinuxNetwork” en een IP address range van 10.0.42.0. Click op OK om de wijzigingen op te slaan.

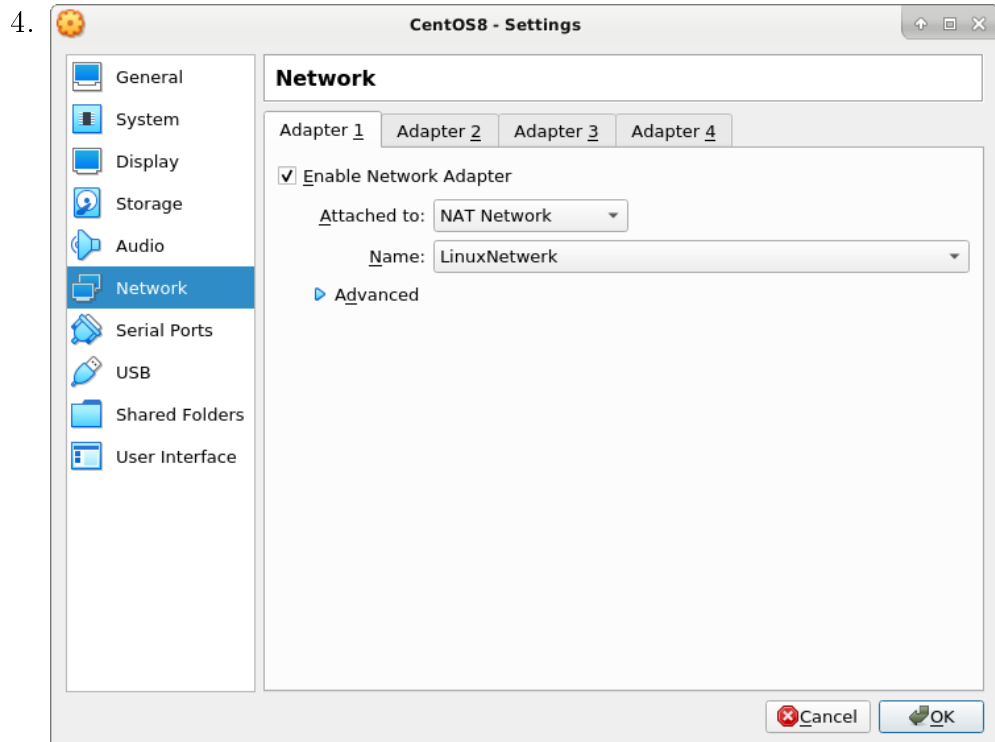


## 5.3 Aanmaken van een Virtual Machine



2. Kies voor een 15 GB harddisk (VDI) die dynamisch mag groeien. Voor het geheugen gebruiken we 2 GB RAM.

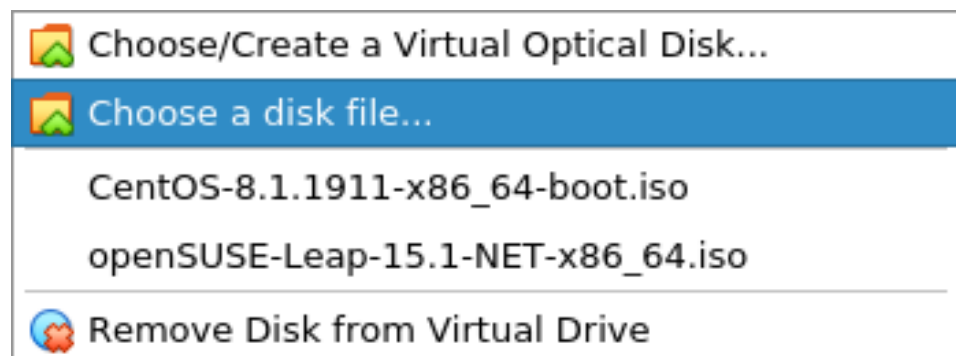
3. Als de machine aangemaakt is gaan we de netwerk settings wijzigen. We selecteren de machine en klikken op Properties. Bij Settings kiezen we voor Network.



Wijzig bij Attached to: de setting naar NAT Network en selecteer vervolgens bij Name: het LinuxNewerk. Click op OK om de gemaakte wijzigingen op te slaan.

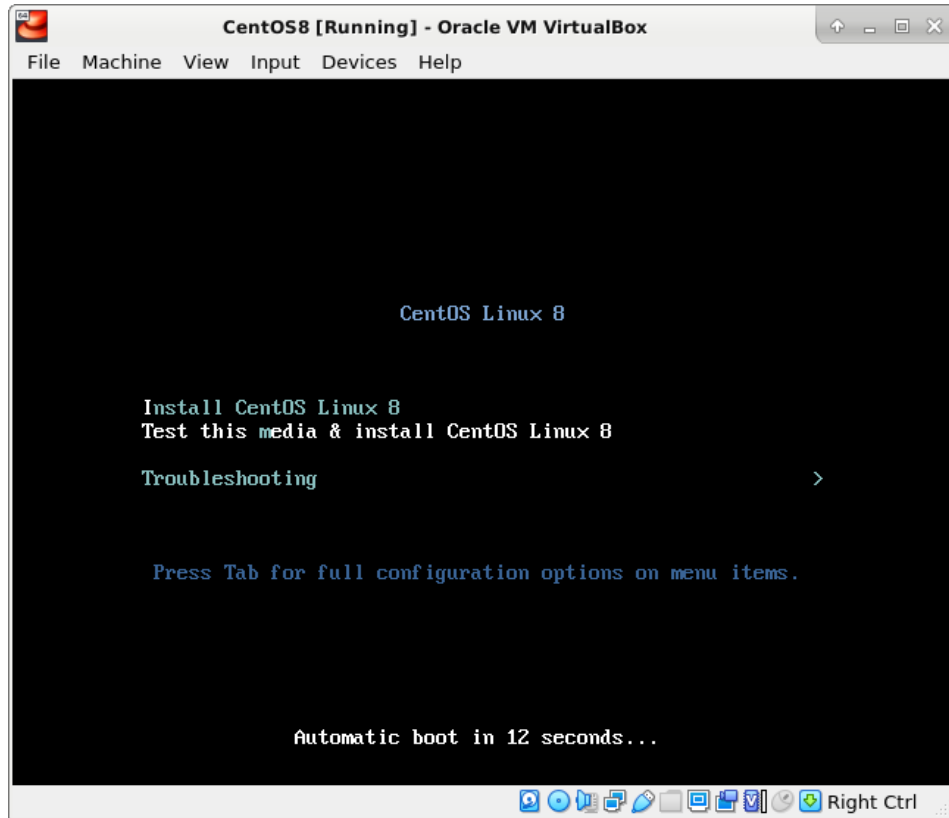
## 5.4 CentOS installatie

1. Verbindt de gedownloade ISO aan de virtuele CD speler door in het Settings menu op storage te cliken en het bestand te koppelen aan de CD.



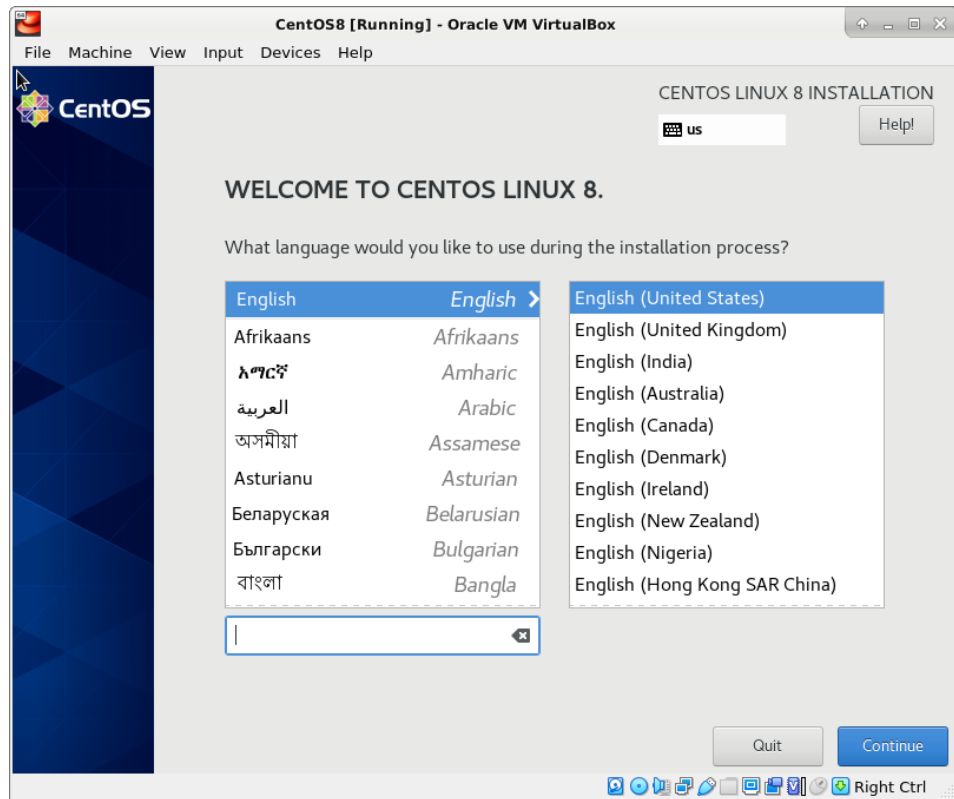
2. Nu mag je de virtuele machine opstarten. Het eerste scherm dat je tegen komt geeft je de mogelijkheid om CentOS te installeren en dat gaan we dan ook doen.

3. Je kan wachten tot de installatie automatisch start of je drukt op enter om de installatie te starten.



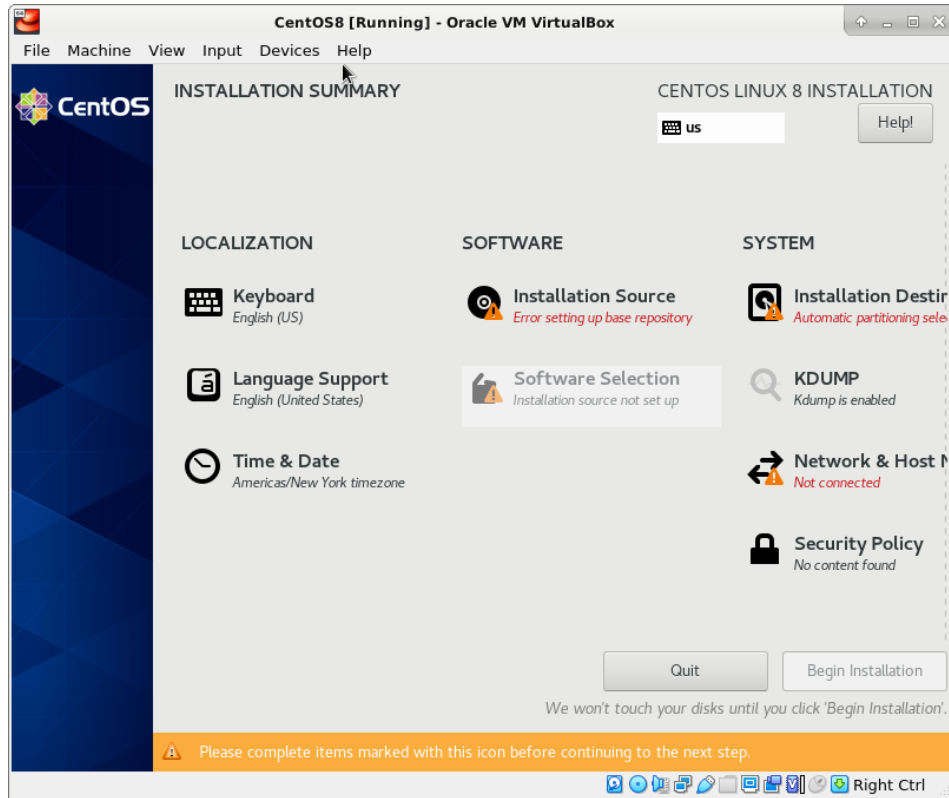
## 22 HOOFDSTUK 5. INSTALLATIE VAN CENTOS OP VIRTUAL BOX

4.



Het is misschien het handigst om een systeem in de Engelse taal te installeren, deze cursus gaat daarvan uit. Het grote voordeel is dat als je iets wil zoeken op het Internet dat je dan al gelijk in het Engels zoekt wat de kans op gelijksoortige problemen groter maakt er zijn immers meer mensen die Engels spreken dan Nederlands. Maar als je niet vertrouwd genoeg vindt in het Engels dan kan je hier ook kiezen voor een Nederlandstalige installatie.

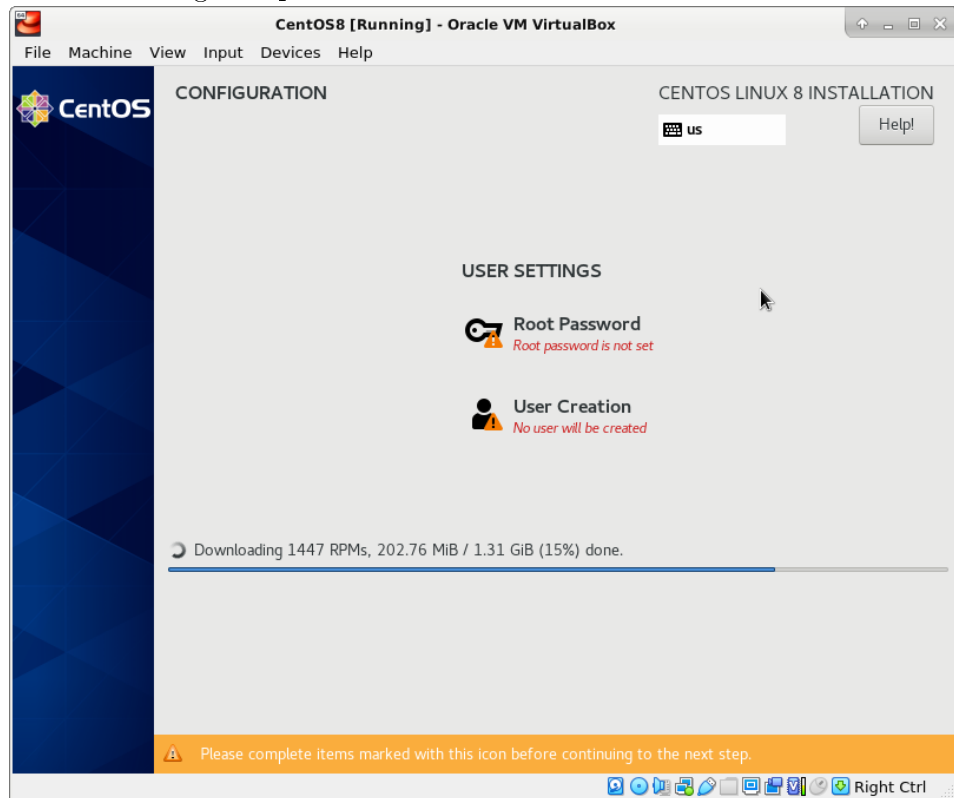
5. In het vervolg scherm heb je heel veel keuzes in 1 keer. We zullen ze in de meest logische volgorde doorlopen.



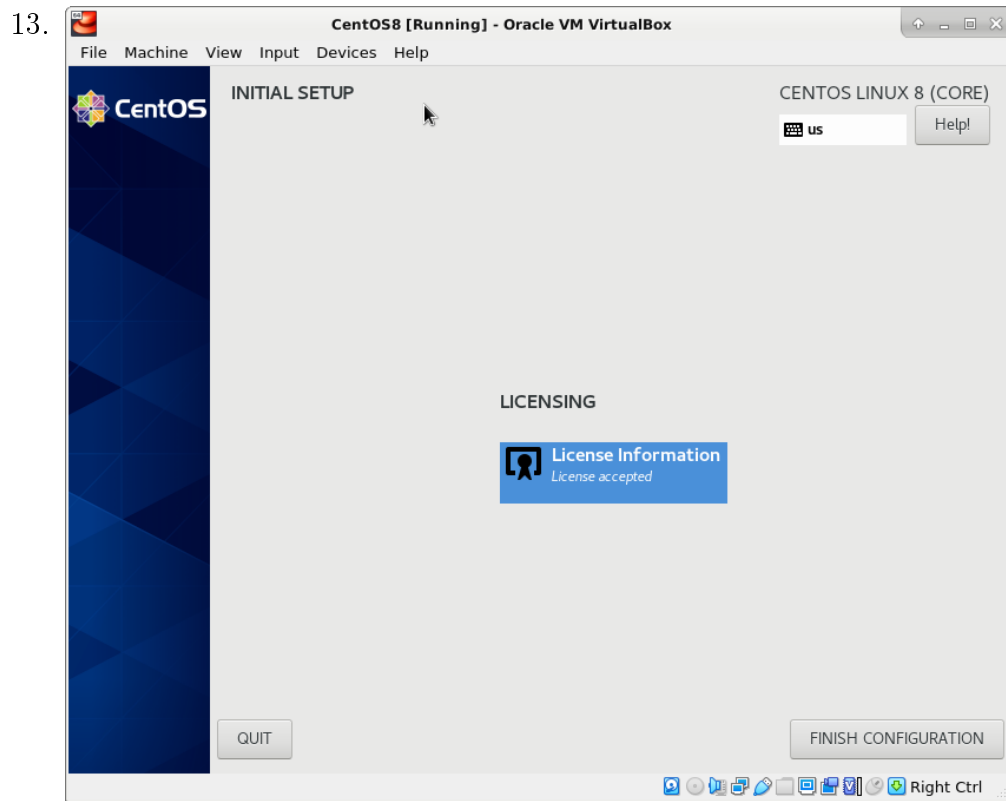
6. We beginnen met Network & Host Name. In het menu dat verschijnt, moet je het netwerk alleen maar op On te zetten en dan op Done te klikken.
7. Vervolgens selecteren we Time & Date en zetten we de tijdzone naar Europe en de plaats Amsterdam. En als dat nog niet aan staat, dan zetten we Network time ook op On. Daarna klikken we weer Done.
8. Nu gaan we voor de Installation Destination optie. Om het makkelijk te houden, gaan we volledig voor de standaard instellingen, dus selecteren we Done.
9. En als laatste kiezen we voor Installation Source. Aan de linkerkant selecteren we Workstation en aan de rechterkant zetten we een vinkje voor Office Suite and Productivity, we sluiten weer af met Done.
10. Nu is de knop Begin Installation donker grijs geworden en kunnen we hem aanklikken.

## 24 HOOFDSTUK 5. INSTALLATIE VAN CENTOS OP VIRTUAL BOX

11. Het systeem zal bezig gaan met de installatie en het geeft ons ondertussen de tijd om een wachtwoord voor root (Administrator van een Linux systeem) in te stellen en een gebruikersaccount voor onszelf te maken. Doe dit alle twee en zorg dat je de wachtwoorden goed onthoud of ergens opslaat.

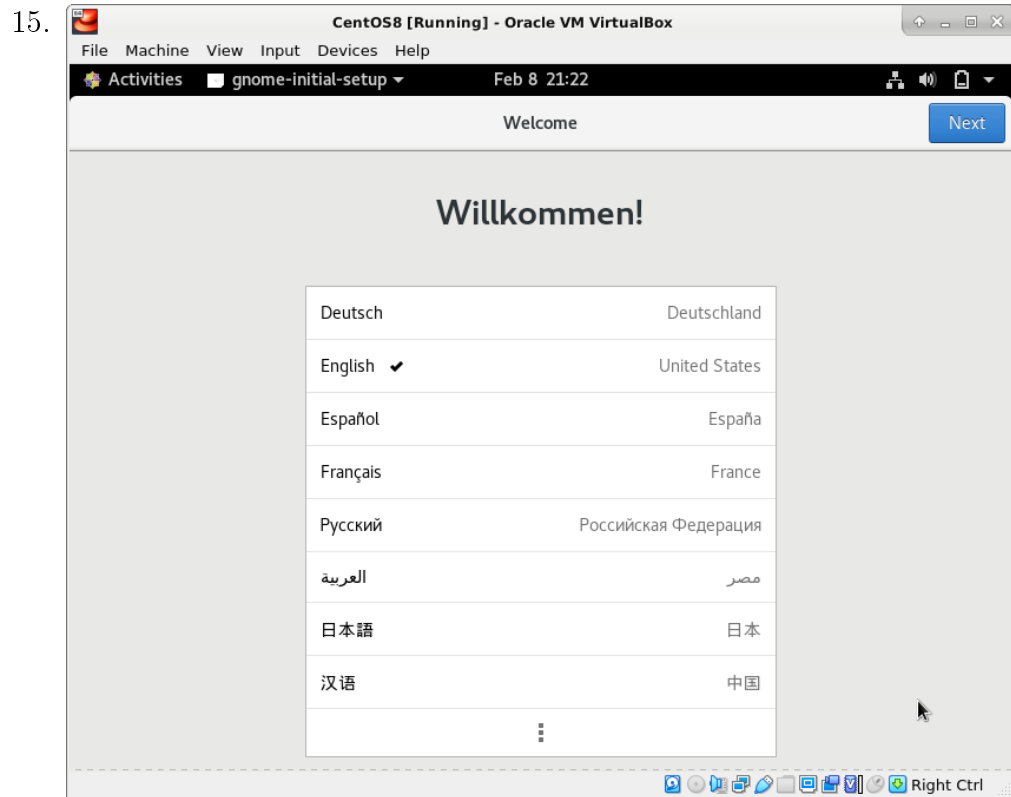


12. Als de installatie klaar is dan moeten we via Settings → Storage de CD uit de virtuele CD speler verwijderen. Daarna kunnen we op Reboot clicken en zal ons nieuwe systeem opstarten.



Na het opstarten en inloggen zal het systeem ons vragen om de licentievoorwaarden te accepteren.

14. Zodra je dit gedaan hebt kan je op FINISH CONFIGURATION cliken en zal de installatie worden afgerond.



Je komt nu in het welkomstscherm terecht.

META Reinstall en zie wat er na Next komt en documenteer dit.



# Index

BSD, <a href="#">4</a>	GNU, <a href="#">4</a>
C, <a href="#">3</a> , <a href="#">7</a>	GPL, <a href="#">4</a>
copyleft, <a href="#">4</a>	Linux, <a href="#">5</a>
End-Users License Agreement, <a href="#">9</a>	POSIX, <a href="#">4</a>
EULA, <a href="#">9</a>	Unix, <a href="#">3</a>
Free Software Foundation, <a href="#">4</a>	