

# Labo-opdracht: installatie van software automatiseren

System Engineering Lab – 2020-2021

*In deze labo-opdracht leer je een techniek aan die je gedurende je studies (en ook daarna) kan helpen bij het installeren en onderhouden van software op je laptop. Meer bepaald leer je gebruik te maken van een package manager om de installatie en onderhoud van software meer gecentraliseerd te beheren. Door het proces te automatiseren met een script, kan je, wanneer je later om de ene of de andere reden je laptop moet herinstalleren, een hoop tijd besparen.*

*Hou een persoonlijk laboverslag bij waarin je jouw resultaten neerschrijft.*

Commented [TA1]: onvolledige zin

## 1 Inhoudsopgave

2	Probleemstelling.....	1
3	Instructies voor Windows-gebruikers .....	2
3.1	Windows PowerShell Integrated Scripting Environment (ISE).....	2
3.2	De Chocolatey Package Manager .....	4
3.3	Het installatiescript afwerken.....	5
4	Instructies voor Mac-gebruikers.....	7
4.1	Een script schrijven .....	7
4.2	De Homebrew package manager.....	9
4.3	Het installatiescript afwerken.....	10
5	Instructies voor Linux-gebruikers .....	11
5.1	Linux als desktop-besturingssysteem .....	11
5.2	Een script schrijven .....	12
5.3	De apt-package manager .....	13
5.4	Het installatiescript afwerken.....	14
6	Evaluatie .....	15

## 2 Probleemstelling

Gedurende je opleiding zal je verschillende softwarepakketten moeten installeren op je laptop. De meeste applicaties zijn *open source* en kan je gratis downloaden en installeren. Zowel Windows- als Mac-gebruikers krijgen dan vaak als instructie om naar de website van de applicatie te gaan, die te downloaden en een installatie-wizard te doorlopen. Dit is een tijdrovende bezigheid. Eens geïnstalleerd,

krijg je bovendien te pas en te onpas notificaties dat er voor de ene of de andere applicatie een update beschikbaar is. Dit zorgt voor onderbrekingen en afleiding op momenten dat je met andere zaken wil bezig zijn.

Linux-gebruikers hebben dit soort problemen niet. Je kan de meeste applicaties die je nodig hebt, zowel voor persoonlijk als professioneel gebruik, downloaden en installeren vanuit een zogenaamde *package manager*. Je kan dit best vergelijken met de App Store, alhoewel het concept op Linux al veel langer bestaat. Je zoekt de applicatie die je wil installeren uit een lijst van duizenden beschikbare *packages* en kan ze installeren met een druk op een knop. Ook het bijwerken van applicaties tot de laatste versie wordt door een *package manager* gedaan. Je kan een lijst opvragen van alle geïnstalleerde applicaties waarvoor een nieuwe versie beschikbaar is en die allemaal in één keer bijwerken.

In deze labo-opdracht zal je een manier leren kennen waarmee je de voordelen van een Linux-package manager naar het Windows- of Mac-platform kan brengen. We gaan meer bepaald een script schrijven dat de meeste applicaties die je nodig hebt voor deze en andere cursussen van de opleiding automatisch installeert.

In de rest van deze labo-opdracht vind je instructies voor de meest gangbare besturingssystemen: **Windows**, **macOS** en **Linux**. Volg de instructies voor het besturingssysteem dat je op je laptop geïnstalleerd hebt. Windows-gebruikers moeten dus bijvoorbeeld niet de instructies voor Linux volgen of omgekeerd.

### 3 Instructies voor Windows-gebruikers

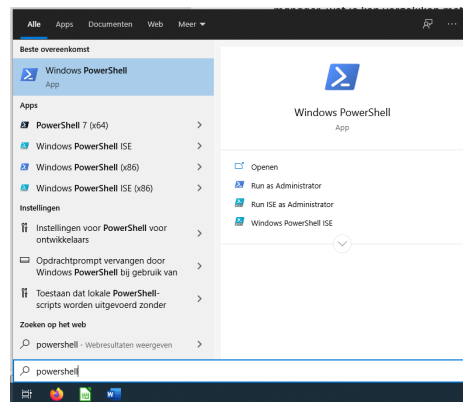
Als je een Windows-gebruiker bent, dan volg je de instructies in deze sectie. Gebruikers van macOS of Linux gaan verder naar resp. Sectie 4 of 5.

#### 3.1 Windows PowerShell Integrated Scripting Environment (ISE)

Windows PowerShell is een Command Line Interface (CLI) voor Windows-systemen. Je kan dit vergelijken met Bash voor Linux. De Windows PowerShell Integrated Scripting Environment (ISE) laat toe om PowerShell-commando's uit te proberen en ook scripts te schrijven, bestanden met CLI-instructies die één na één uitgevoerd worden en die toelaten om tijdrovende systeembeheertaken (zoals het installeren van software) te automatiseren. De terminal lijkt op het eerste gezicht een verouderde manier om van een computer iets gedaan te krijgen als je even goed kan gebruik maken van een grafische applicatie. De CLI en scripts zijn echter de sleutel tot automatisering en zijn essentieel bij het beheren van computersystemen op grote schaal.

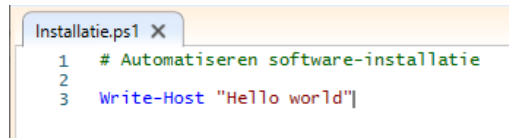
1. Druk op de Windows-toets of klik in het zoekveld naast het Windows-icoon linksonder het scherm. Tik "PowerShell" in, en klik op "Run ISE as Administrator" (zie Figuur 1). Bevestig dat de ISE mag veranderingen aanbrengen aan je computer.
2. In PowerShell ISE heb je drie grote onderdelen: een script-editor, een PowerShell console en een lijst met PowerShell-commando's (zie Figuur 2). We gaan een eerste eenvoudig script schrijven. Tik in het editor-venster de tekst in zoals getoond in Figuur 3 en sla het script op als

3. Zoek in de knoppenbalk bovenaan het editor-venster naar de knop waarmee je het script kan uitvoeren (deze ziet eruit als een groene driehoek) en klik erop, of druk op de functietoets F5. Controleer dat het script uitgevoerd wordt in het consolevenster (zie Figuur 4).



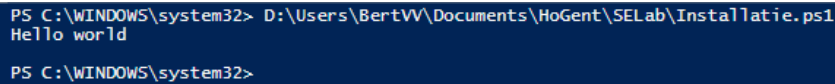
The screenshot displays a Windows 10 desktop environment. In the background, the Windows Defender Security Center application is open, showing the 'Virus & threat protection' section. In the foreground, a Windows PowerShell window is active, displaying the command 'Get-NetFirewallPolicy' and its output. The output lists various firewall policies, including 'All', 'Domain', 'Private', and 'Public', and their associated settings. To the right, the Windows Defender Firewall control panel is open, showing the 'Advanced settings' tab. The 'Inbound rules' section is expanded, showing a list of rules, including 'Windows Defender Firewall' and 'Windows Defender Firewall with Internet Explorer'. The 'Outbound rules' section is also expanded, showing a list of rules, including 'Windows Defender Firewall' and 'Windows Defender Firewall with Internet Explorer'.

*Figuur 2. Windows PowerShell ISE. Linksonder zie je een editor voor het schrijven van scripts. Onderaan bevindt zich een PowerShell-console waar je meteen commando's in kan uitvoeren. Rechts zie je een lijst van PowerShell-commando's die je kan doorzoeken.*



```
Installatie.ps1 X
1 # Automatiseren software-installatie
2
3 Write-Host "Hello world"
```

Figuur 3. Ons eerste PowerShell-script. De eerste regel die begint met een hekje (#) is commentaar en wordt niet uitgevoerd. Het commando op regel 3, `Write-Host`, drukt tekst af op de console.



```
PS C:\WINDOWS\system32> D:\Users\BertVW\Documents\HoGent\SELab\Installatie.ps1
Hello world

PS C:\WINDOWS\system32>
```

Figuur 4. Resultaat van het uitvoeren van het script.

Beantwoord nu onderstaande vragen:

- De PowerShell-prompt toont de map/directory waar we ons nu bevinden. Wat is de naam van deze directory?
- In welke directory heb je het script opgeslagen?
- In welke directory is het script opgeslagen in de screenshot in Figuur 4?

Voer ook onderstaande opdrachten uit:

- Gebruik het commando “`cd`” (change directory) om naar de directory te gaan waar je je script hebt opgeslagen. Merk op dat de PowerShell-console je helpt om de naam van de directory te vervullen. Je kan ook de TAB-toets gebruiken om een fragment van een directorynaam te vervullen.
- In de directory waar je het script hebt opgeslagen, tik je de eerste letters van de naam (bv. “`Ins`”) in en druk je op TAB. Merk op dat de PowerShell-console dit vervolledigt tot “`.\Installatie.ps1`”. Druk ENTER en controleer of het script correct uitgevoerd wordt.
- (optioneel) Verander de kleuren van de uitvoer, bv. Groene tekst op zwarte achtergrond. Tip: gebruik de commandolijst rechts om de juiste syntax te bepalen.

### 3.2 De Chocolatey Package Manager

We laten ISE even voor wat het is. Sluit het nog niet af want zo dadelijk heb je het opnieuw nodig!

In Windows is er een App Store (de Microsoft Store), maar daar vind je vooral commerciële software of spelletjes. In dit labo gaan we aan de slag met *Chocolatey*, een package manager die is ontwikkeld buiten Microsoft. Microsoft is intussen ook bezig met de ontwikkeling van een gelijkaardig systeem, WinGet<sup>1</sup>, maar dat is op dit moment nog niet officieel uitgekomen.

1. Ga naar de Chocolatey-webpagina (<https://chocolatey.org/>) en lees de algemene informatie (o.a. *Why Chocolatey*, *How Chocolatey Works*). Heb je het moeilijk met het lezen van Engels?

<sup>1</sup> Zie <https://docs.microsoft.com/en-us/windows/package-manager/winget/>

Sommige browsers laten toe om een website te vertalen. Je kan ook tekst kopiëren en plakken in Google Translate.

2. Zoek de instructies voor het installeren van Chocolatey en lees ze (desnoods door middel van vertaalsoftware).
3. Kopieer de volgende opdrachtregel om Chocolatey te installeren in de console van ISE en druk ENTER:  
`Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))`
4. Normaal is Chocolatey nu geïnstalleerd. Lees de uitvoer van het installatieproces uit de vorige stap door en controleer of er geen fatale fouten gebeurd zijn. Controleer of Chocolatey correct geïnstalleerd is door het commando “choco -?” uit te voeren in de PowerShell ISE console. Als dit lukt, krijg je een overzicht van de opties en parameters die je kan gebruiken met het commando choco.
5. Om een applicatie te installeren gebruik je het commando “choco install <packagenaam>”. Probeer dit eens uit. Voor het programmeerproject zullen jullie gebruik maken van het versiebeheersysteem “Git”. Installeer dit met het commando: “choco install git”. Merk op dat Chocolatey je wellicht vraagt te bevestigen of je Git wel degelijk wilt installeren. In een script willen we vermijden dat we om de paar minuten iets moet intikken. Daar zullen we rekening moeten mee houden bij het schrijven van het installatiescript.

#### Bijkomende opdrachten

Op dit punt wordt het interessant om een lijstje op te bouwen van commando's die je vaak zult nodig hebben. Zoek op (in de handleiding van Chocolatey of gewoon met Google) welke commando's je nodig hebt voor de hieronder opgesomde taken en hou dit bij op een plaats waar je dit in de toekomst makkelijk terugvindt. Voeg het zeker ook toe in je laboverslag.

Taak	Commando
Een lijst tonen van de software die nu geïnstalleerd is via Chocolatey	
Alle packages die nu geïnstalleerd zijn bijwerken tot de laatste versie	
Via de console een package opzoeken	
Een geïnstalleerde applicatie verwijderen	

### 3.3 Het installatiescript afwerken

We weten nu dat je een applicatie kan installeren met “choco install <packagenaam>”. De exacte naam van de package kan je opzoeken via de website (<https://chocolatey.org/packages/>). Elke applicatie die je moet installeren kan je nu toevoegen aan het script, zodat je dit kan automatiseren. Pas het script van daarstraks aan:

```
# Automatiseren software-installatie
```

```
write-Host "Installatie algemene applicaties"
choco install -y git
```

De optie -y zorgt ervoor dat je tijdens de uitvoering van het script niet meer moet bevestigen dat je Git wel wil installeren. Merk op dat als een applicatie al geïnstalleerd is, je een waarschuwing krijgt, maar het script wel kan verder lopen. Het kan dus geen kwaad om het script verschillende keren na elkaar uit te voeren.

Zoek de naam op van de packages voor de hieronder opgesomde applicaties, en vul je installatiescript aan. Voeg telkens een regel toe die begint met “choco install -y” gevolgd door de naam van de package. Als je de applicatie al geïnstalleerd hebt, kan je overwegen om deze even te verwijderen en opnieuw te installeren via Chocolatey. Dat zal later het up-to-date houden van je systeem vereenvoudigen, want dan kan je choco gebruiken om alle software in één keer bij te werken.

- Adobe Acrobat Reader (of een andere applicatie om PDF's te openen, bv FoxIt Reader)
- Firefox: de bekende webbrowser (of jouw favoriete webbrowser, voor zover die beschikbaar is via Chocolatey)
- GitHub Desktop: een grafische applicatie voor het Git versiebeheersysteem
- Visual Studio Code: een krachtige teksteditor met syntaxkleuren en ondersteuning voor verschillende programmeer-, scripting- en markuptalen.
- VLC Media Player

Voor System Engineering Lab heb je bovendien nog volgende applicaties nodig:

- FileZilla: kopiëren van bestanden van/naar servers (via FTP, SFTP, enz.)
- VirtualBox: laat toe om zgn. virtuele computers aan te maken, op te starten en te installeren met een besturingssysteem
- MySQL Workbench

Je kan zelf het installatiescript onderverdelen in “rubrieken”, bv. software die je voor verschillende vakken nodig hebt kunnen onder de lijn die “Installatie algemene applicaties” op de console afdrukt. Software die je nodig hebt voor een specifiek vak kan je onder een aparte hoofding plaatsen, bv.

```
write-Host "Software voor System Engineering Lab"
```

Structureer zelf je script en zorg er eventueel voor dat deze hoofdingen meer in het oog springen, bijvoorbeeld door ze in kleur of met een kadertje er rond af te drukken. Je kan je script ook uitbreiden met applicaties die je voor andere vakken nodig hebt of die je zelf gebruikt, maar die niet noodzakelijk nodig zijn voor de opleiding. Zoek eens in de lijst van beschikbare packages naar nuttige applicaties.

Merk op dat je commerciële software zoals Cisco PacketTracer of VisualParadigm niet via Chocolatey kan installeren. Dit soort applicaties zal je nog altijd op de “traditionele” manier moeten installeren...

Als laatste stap in het labo probeer je het commando uit om alle geïnstalleerde applicaties bij te werken tot de laatste versie. Natuurlijk zal dat nu geen effect hebben, maar het is nuttig het alvast eens uit te proberen!

Indien het niet lukt, kan je een Panopto opname over dit labo bekijken op

<https://hogent.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=0c1b2544-04b4-415b-ac97-acc80170b9e3>.

## 4 Instructies voor Mac-gebruikers

Als je een Mac-gebruiker bent, dan volg je de instructies in deze sectie. Gebruikers van Windows of Linux gaan naar resp. Sectie 3 of 5.

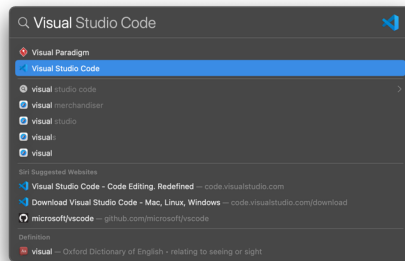
### 4.1 Een script schrijven

Elk macOS-systeem is, net als Linux, uitgerust met een krachtige command-line interface (CLI), toegankelijk via de **Terminal** applicatie. De standaard shell op macOS is **Zsh**, maar ook **Bash** — wat je al kent van Linux — is beschikbaar.

Als je een grafische gebruikersomgeving gewend bent, lijkt een terminal op het eerste gezicht een totaal verouderde manier om van een computer iets gedaan te krijgen. De CLI en **scripts** (bestanden met CLI-instructies die één na één uitgevoerd worden) zijn echter essentieel bij het beheren van computersystemen op grote schaal. Ze laten je toe om tijdrovende systeembeheertaken (zoals het installeren van software) te automatiseren.

In de volgende stappen ga je je eerste script aanmaken:

1. Open een teksteditor zoals TextEdit of Visual Studio Code. Dit kan bvb. met de sneltoets **Cmd + Spatie** (zie Figuur 5).
2. Maak een nieuw tekstbestand aan en geef het de inhoud die je ziet in Figuur 6. Sla dit bestand op als **installatie.sh** (de extensie **.sh** is standaard voor scripts). Kies zelf een geschikte map zodat je het script later makkelijk kan terugvinden.
3. Open Terminal en gebruik het commando **cd** om te navigeren naar de directory waar je het script hebt opgeslagen. Gebruik het commando **ls** om te verifiëren dat het script zich wel degelijk in deze directory bevindt.
4. Voer het volgende commando uit om van je tekstbestand een uitvoerbaar script te maken:  
`chmod +x installatie.sh`
5. Voer het script als volgt uit:  
`./installatie.sh`
6. Controleer dat het resultaat van het script overeenkomt met Figuur 7.



Figuur 5. Een applicatie openen via Cmd + Spatie. In dit voorbeeld wordt Visual Studio Code opgestart.

```
1  #!/bin/bash
2  # Automatiseren software-installatie
3
4  echo "Hello world!"
5
```

Figuur 6. Ons eerste shell script. De eerste regel begint met `#!/`, de zogenaamde "shebang". Die geeft aan met welke shell het script moet uitgevoerd worden (hier: Bash). De tweede regel begint met een hekje (`#`); dit is commentaar die niet uitgevoerd wordt. Regel 4 toont tekst in de console.

```
steven@Stevens-MBP ~ % cd Documents/SysEngLab
steven@Stevens-MBP SysEngLab % ls
installatie.sh
steven@Stevens-MBP SysEngLab % chmod +x installatie.sh
steven@Stevens-MBP SysEngLab % ./installatie.sh
Hello world!
steven@Stevens-MBP SysEngLab %
```

Figuur 7. Resultaat van het uitvoeren van het script.

Beantwoord nu onderstaande vragen:

- De shell prompt toont telkens de directory waarin je je bevindt. In welke directory kom je terecht als je een Terminal-venster opent?



- In welke directory heb jij het script opgeslagen?
- In welke directory is het script opgeslagen in de screenshot in Figuur 7?

## 4.2 De Homebrew package manager

We laten het script even voor wat het is. Sluit de editor nog niet af want zo dadelijk heb je die opnieuw nodig!

Standaard kan je op macOS enkel programma's installeren via de App Store (zoals op iOS) of handmatig; er is geen package manager geïnstalleerd. De App Store bevat vooral commerciële software en games. Wij gaan in deze labo's gebruik maken van **Homebrew**, een package manager die ontwikkeld is buiten Apple.

Voer de volgende stappen uit:

1. Ga naar <https://brew.sh> en lees de algemene informatie over Homebrew (o.a. *What Does Homebrew Do?*). Heb je het moeilijk met het lezen van Engels? Er is ook een Nederlandstalige versie beschikbaar, die kan je selecteren onder het zoekveld.
2. Zoek de instructies voor het installeren van Homebrew en voer ze uit door het installatiecommando (dat begint met `/bin/bash`) uit te voeren in Terminal.
3. Lees de uitvoer van het installatieproces en controleer of er geen fatale fouten gebeurd zijn.
4. Controleer of Homebrew correct geïnstalleerd is door het commando `brew --help` uit te voeren. Als dit lukt, dan krijg je een overzicht van de opties en parameters die je kan gebruiken met het commando **brew**.
5. Om een applicatie te installeren gebruik je het volgende commando:  
`brew install <packagenaam>`
6. Voor het programmeerproject zullen jullie gebruik maken van het versiebeheersysteem **Git**. Installeer dit met het commando `brew install git`.

### Bijkomende opdrachten

Op dit punt wordt het interessant om een lijstje op te bouwen van commando's die je vaak zult nodig hebben. Zoek op (in de handleiding van Homebrew of gewoon met Google) welke commando's je nodig hebt voor de hieronder opgesomde taken en hou dit bij op een plaats waar je dit in de toekomst makkelijk terugvindt. Voeg ze zeker ook toe aan je laboverslag.

Taak	Commando
Een lijst tonen van de software die nu geïnstalleerd is via Homebrew	
Alle packages die nu geïnstalleerd zijn bijwerken tot de laatste versie	
Via de console een package opzoeken	
Een geïnstalleerde applicatie verwijderen	

### 4.3 Het installatiescript afwerken

Je weet nu hoe je een applicatie kan installeren met Homebrew. Elke applicatie die je moet installeren kan je nu toevoegen aan je script, zodat je dit kan automatiseren. Pas het script van daarstraks aan:

```
#!/bin/bash
# Automatiseren software-installatie

echo "Installatie algemene applicaties"
brew install git
```

Merk op dat als een applicatie al geïnstalleerd is, je een waarschuwing krijgt, maar het script wel kan verder lopen. Het kan dus geen kwaad om het script verschillende keren na elkaar uit te voeren.

De package naam die hoort bij een applicatie kan je opzoeken via de website <https://formulae.brew.sh>. Merk op dat Homebrew onderscheid maakt tussen command-line applicaties en GUI applicaties (“apps”). Deze laatste heten **casks** en vereisen de optie **--cask** in je installatiecommando.

Zoek de naam op van de packages voor de hieronder opgesomde applicaties, en vul je installatiescript aan met de installatiecommando’s voor deze applicaties:

- Firefox: de bekende webbrowser (of kies een andere webbrowser).
- GitHub Desktop: een grafische applicatie voor het Git versiebeheersysteem
- Visual Studio Code: een krachtige teksteditor met syntaxkleuren en ondersteuning voor verschillende programmeer-, scripting- en markuptalen.
- VLC Media Player
- Visual Paradigm
- Microsoft Teams

Voor System Engineering Lab heb je bovendien nog volgende applicaties nodig:

- FileZilla: kopiëren van bestanden van/naar servers (via FTP, SFTP, enz.).
- VirtualBox: laat toe om zgn. virtuele computers aan te maken, op te starten en te installeren met een besturingssysteem.
- MySQL Workbench

Voeg deze applicaties ook toe aan je script.

Je kan zelf het installatiescript onderverdelen in “rubrieken”, bv. software die je voor verschillende vakken nodig hebt kunnen onder de lijn die “Installatie algemene applicaties” op de console afdrukt. Software die je nodig hebt voor een specifiek vak kan je onder een aparte hoofding plaatsen, bv.

```
echo "Software voor System Engineering Lab"
```

Je kan je script ook uitbreiden met applicaties die je voor andere vakken nodig hebt of die je zelf gebruikt, maar die niet noodzakelijk nodig zijn voor de opleiding. Zoek eens in de lijst van beschikbare packages naar nuttige applicaties.

Merk op dat je niet alle commerciële software, zoals bv. Cisco PacketTracer, via Homebrew kan installeren. Dit soort applicaties zal je nog altijd op de “traditionele” manier moeten installeren...

Als laatste stap in het labo probeer je het commando uit om alle geïnstalleerde applicaties bij te werken tot de laatste versie. Natuurlijk zal dat nu geen effect hebben, maar het is nuttig om het alvast eens uit te proberen!

## 5 Instructies voor Linux-gebruikers

Als je een Linux-gebruiker bent, dan volg je de instructies in deze sectie. Gebruikers van Windows of macOS gaan naar resp. Sectie 3 of 4.

### 5.1 Linux als desktop-besturingssysteem

Onder de studenten toegepaste informatica zijn de Linux-gebruikers meestal een kleine minderheid. Dit is te begrijpen, want als je een laptop koopt staat er meestal al Windows op. Bovendien wordt er op HOGENT vanuit de IT-helpdesk geen ondersteuning geboden aan Linux-gebruikers. Desondanks is Linux gebruiken zeker het overwegen waard. Voor de meeste IT-professionals is Linux een uitstekend besturingssysteem. Vele applicaties voor softwareontwikkeling, systeem- en netwerkbeheer, cybersecurity, enz. zijn gratis beschikbaar. Ook voor bekende commerciële applicaties zoals MS Office zijn er compatibele alternatieven beschikbaar (bv. LibreOffice). Qua gebruiksvriendelijkheid zijn de moderne Linux-distributies te vergelijken met macOS. Enkel wie die gebonden is aan bepaalde commerciële applicaties die alleen maar op Windows en/of macOS beschikbaar zijn, heeft geen reden om Linux te overwegen: ontwikkelaars die specifiek software schrijven voor het Windows- of het Mac/iOS-platform, Windows systeembeheerders, enz.

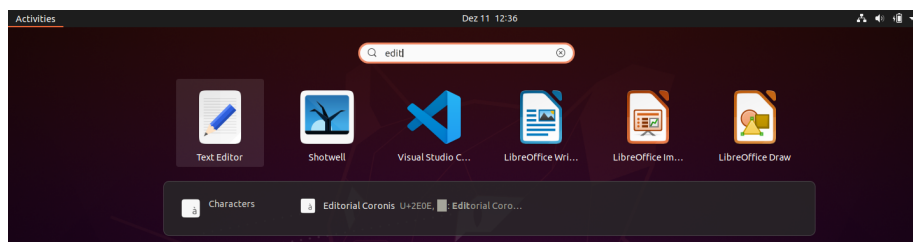
Eigenlijk is Linux enkel de “kern” van het besturingssysteem en heb je daarnaast nog een hoop programmabibliotheken, grafische gebruikersomgeving, tools en applicaties nodig om een volwaardig systeem te vormen. Verschillende organisaties en bedrijven hebben hun eigen versie van Linux uitgebracht met een selectie van tools en applicaties. Dit wordt een *distributie* genoemd. Enkele bekende voorbeelden zijn Ubuntu, Debian, RedHat, Fedora, Arch, enz. Het is jammer genoeg niet eenvoudig om eenduidige instructies voor Linux-gebruikers te formuleren. Eén van de verschillen tussen de vele distributies is immers vaak de *package manager*. In wat hieronder volgt, veronderstellen we dat je **Ubuntu 20.04 LTS** gebruikt, omdat dat een bekende, stabiele distributie is die ook geschikt is voor beginnende gebruikers. De instructies zullen hoogstwaarschijnlijk ook zonder wijzigingen werken voor Debian, Linux Mint of gelijkaardige distributies uit de Debian-familie. Heb je toch een andere distributie geïnstalleerd? Contacteer Bert Van Vreckem ([bert.vanvreckem@hogent.be](mailto:bert.vanvreckem@hogent.be)) voor specifieke hulp. Geef zeker mee welke distributie en versie je geïnstalleerd hebt op je laptop.

## 5.2 Een script schrijven

Elk Linux-systeem is uitgerust met een krachtige Command Line Interface (CLI), genaamd Bash. De naam is een afkorting van de Bourne Again Shell, genoemd naar Stephen Bourne, de oorspronkelijke auteur. Als je een grafische gebruikersomgeving gewend bent, lijkt de terminal op het eerste gezicht een totaal verouderde manier om van een computer iets gedaan te krijgen. De CLI en scripts (bestanden met CLI-instructies die één na één uitgevoerd worden) zijn echter de sleutel tot automatisering en zijn essentieel bij het beheren van computersystemen op grote schaal.

Er is ook een grote keuze aan applicaties om tekst te bewerken. In dit labo gebruiken we de standaard teksteditor, GEdit, om een script te schrijven dat de installatie van software die je nodig hebt te automatiseren. Als je wilt kan je zelf een andere teksteditor kiezen.

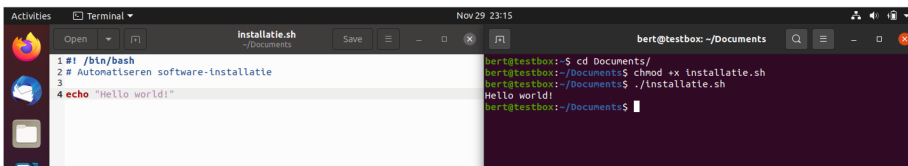
1. Druk op de Windows-toets of klik bovenaan op “Activities” en tik “Edit”. Klik in de lijst met resultaten op “Text Editor” of druk ENTER als het icoontje geselecteerd is (zie Figuur 8).
2. Druk opnieuw op de Windows-toets, tik “Terminal” en druk ENTER. Zorg ervoor dat de teksteditor en de terminal elk de helft van het scherm innemen door de titelbalk naar hetzij de linker- hetzij de rechterrاند van het scherm te slepen (zie Figuur 10). Je kan ook de toetsenbordcombinatie Windows + pijltje links of pijltje rechts gebruiken.
3. We gaan een eerste eenvoudig script schrijven. Tik in het editor-venster de tekst in zoals getoond in Figuur 9 en sla het script op als “installatie.sh”. De extensie .sh is standaard voor Bash-scripts. Kies zelf een geschikte map om het script op te slaan zodat je het later makkelijk kan terugvinden.
4. Ga in het terminalvenster met `cd` naar de directory waar je het script hebt opgeslagen. Controleer met `ls` dat het script zich wel degelijk in deze directory bevindt. Voer het commando `“chmod +x installatie.sh”` uit. Dit zorgt ervoor dat het script uitvoerbaar wordt. Voer het script uit met `“./installatie.sh”` en controleer dat het resultaat overeenkomt met Figuur 10.



Figuur 8. Om een applicatie te openen, klik linksboven op Activities en tik een deel van de naam van de gewenste applicatie in. Hier wordt de standaard teksteditor (GEdit) opgestart.

```
1 #! /bin/bash
2 # Automatiseren software-installatie
3
4 echo "Hello world!"
```

Figuur 9. Ons eerste shell-script. De eerste regel begint met "#!", de zogenaamde "shebang". Die geeft aan met welke interpreter het script moet uitgevoerd worden (hier: Bash). De tweede regel begint met een hekje (#), dit is commentaar die niet uitgevoerd wordt. Regel 4 drukt tekst af op de console.



Figuur 10. Resultaat van het uitvoeren van het script.

- De Bash-prompt toont de map/directory waar we ons nu bevinden. Wat is de naam van de directory waar je in terecht komt als je een Terminal-venster opent?
- In welke directory heb jij het script opgeslagen?
- In welke directory is het script opgeslagen in de screenshot in Figuur 10?
- (optioneel) Verander de kleuren van de uitvoer, bv. gele tekst op zwarte achtergrond. Tip: zoek op het internet naar "Bash Ansi color codes" om te weten te komen hoe dit kan.

### 5.3 De apt-package manager

We laten het script even voor wat het is. Sluit de editor nog niet af want zo dadelijk heb je die opnieuw nodig!

Elke Linux-distributie heeft een package manager die toelaat om in een lijst van duizenden beschikbare packages op te zoeken welke je nodig hebt en die met een druk op de knop te installeren. Deze package manager kan je ook altijd vanop de terminal gebruiken. In Ubuntu gebruik je in de Terminal het commando `apt` om de package manager aan te spreken. Om een applicatie te installeren gebruik je het commando `apt install <packagenaam>`. Voor elk commando dat wijzigingen aan het systeem aanbrengt heb je root-rechten nodig. Deze bekom je door het commando te laten voorafgaan door `sudo`. De eerste keer dat je dit gebruikt in een sessie moet je het wachtwoord van je huidige gebruiker invoeren. Probeer dit eens uit. Voor het programmeerproject zullen jullie gebruik maken van het versiebeheersysteem "Git". Installeer dit met het commando: `sudo apt install git`. Merk op dat apt je wellicht vraagt te bevestigen of je Git wel degelijk wilt installeren. In een script willen we vermijden dat we om de paar minuten iets moet intikken. Daar zullen we rekening moeten mee houden bij het schrijven van het installatiescript.

Op dit punt wordt het interessant om een lijstje op te bouwen van commando's die je vaak zult nodig hebben. Zoek op welke commando's je nodig hebt voor de hieronder opgesomde taken en hou dit bij op een plaats waar je dit in de toekomst makkelijk terugvindt. Voeg het in elk geval toe aan je laboverslag.

Taak	Commando
Een lijst tonen van de software die nu geïnstalleerd is via apt	
Alle packages die nu geïnstalleerd zijn bijwerken tot de laatste versie	
Via de console een package opzoeken	
Een geïnstalleerde applicatie verwijderen	

## 5.4 Het installatiescript afwerken

We weten nu dat je een applicatie kan installeren met “`sudo apt install <packagenaam>`”. Pas het script van daarstraks nu aan:

```
1 #! /bin/bash
2 # Automatiseren software-installatie
3
4 echo "=== Bijwerken lijst beschikbare applicaties ==="
5 sudo apt -y update
6
7 echo "=== Algemene applicaties ==="
8 sudo apt -y install git
```

De optie `-y` zorgt ervoor dat je tijdens de uitvoering van het script niet meer moet bevestigen dat je Git wel wil installeren. Je kan in principe een willekeurig aantal packages tegelijk installeren door de namen toe te voegen aan het commando dat nu enkel Git installeert, gescheiden door spaties. Merk op dat als een applicatie al geïnstalleerd is, je een waarschuwing krijgt, maar het script wel kan verder lopen. Het kan dus geen kwaad om het script verschillende keren na elkaar uit te voeren.

Vele applicaties die je op Windows steevast zelf moet installeren zijn al standaard geïnstalleerd op Linux: een PDF-viewer, webbrowser, mediaspeler, enz. De volgende applicaties zijn wellicht niet standaard geïnstalleerd, maar zijn wel nuttig:

- GitKraken: een grafische applicatie voor het Git versiebeheersysteem (Github desktop werkt niet op Linux)
- Visual Studio Code: een krachtige teksteditor met syntaxkleuren en ondersteuning voor verschillende programmeer-, scripting- en markuptalen.
- VLC Media Player

Let op! Visual Studio Code en GitKraken zitten niet in de apt-package repository, maar kan je installeren via “snap”, een alternatieve package manager. Waarom er meer dan één package manager is en wat het verschil is tussen Apt en Snap is iets waar we nu niet kunnen op ingaan, maar je kan dit zelf opzoeken als je het je afvraagt. Het juiste commando is:

```
sudo snap install --classic code
sudo snap install --classic gitkraken
```

Voor System Engineering Lab heb je volgende applicaties nodig:

- FileZilla: kopiëren van bestanden van/naar servers (via FTP, SFTP, enz.)
- VirtualBox: laat toe om zgn. virtuele computers aan te maken, op te starten en te installeren met een besturingssysteem
- MySQL Workbench

Je kan zelf het installatiescript onderverdelen in “rubrieken”, bv. software die je voor verschillende vakken nodig hebt kunnen onder de lijn die “Installatie algemene applicaties” op de console afdrukt. Software die je nodig hebt voor een specifiek vak kan je onder een aparte hoofding plaatsen, bv.

```
echo "=== Software voor System Engineering Lab ==="
```

Structureer zelf je script en zorg er eventueel voor dat deze hoofdingen meer in het oog springen, bijvoorbeeld door ze in kleur of met een kadertje er rond af te drukken. Je kan je script ook uitbreiden met applicaties die je voor andere vakken nodig hebt of die je zelf gebruikt, maar die niet noodzakelijk nodig zijn voor de opleiding. Zoek eens in de lijst van beschikbare packages naar nuttige applicaties.

Merk op dat je commerciële software zoals Cisco PacketTracer of Visual Paradigm niet via Chocolatey kan installeren. Dit soort applicaties zal je nog altijd op de “traditionele” manier moeten installeren...

## 6 Evaluatie

Als je het labo volledig hebt uitgevoerd, toon je het resultaat aan je begeleider. Je hebt dit labo naar behoren uitgevoerd als je:

- Een voldoende gedetailleerd laboverslag geschreven hebt met o.a. het antwoord op vragen gesteld in deze opgave en andere nuttige informatie (screenshots, enz.)
- Chocolatey (Windows) of Homebrew (macOS) geïnstalleerd hebt
- Een script (PowerShell of Bash, afhankelijk van het besturingssysteem op je laptop) geschreven en gebruikt hebt om de opgesomde applicaties te installeren

Mogelijke uitbreidingsdoelen:

- Het script structureren per cursus
- Het script aanvullen met andere software, bv. voor andere vakken of voor eigen gebruik