

Operating Systems

Praktijk - Processen

Doel

Tijdens dit labo zullen we aan de hand van eenvoudige commando's de werking van processen en hun bijhorende informatie achterhalen op zowel een Linux als Windows besturingssysteem. De commando's in Linux zullen uitgevoerd worden in een Bash terminal, deze voor Windows via een PowerShell console.

Na dit labo kan je:

- Een overzicht van alle processen opvragen
- Een overzicht van alle processen van de gebruiker opvragen
- Een detailoverzicht voor één proces opvragen
- Een proces starten, pauzeren en stoppen

Verloop

Tekst in een grijze kader zijn commando's die je zal uitvoeren in de console van het respectievelijke besturingssystemen. Indien de kader leeg is verwachten we dat *jij* deze aanvult met jouw uitgevoerde commando('s).

Voor dit labo neem je de tekst stap voor stap door en neem je de commando's over indien gegeven in de bijhorende console. Aangezien in Powershell niet alle commando's standaard aanwezig zijn, zal je soms door ons aangeleverde scriptjes (zie Chamilo) moeten uitvoeren ter vervanging van een commando.

Doorheen de tekst en in de commando's zal je vaak tekst tussen < > zien staan. Dit betekent dat je deze moet vervangen door een passende waarde voor jouw situatie. Bijvoorbeeld: `ps <procesid>` wordt voor een proces met id 15 `ps 15`.

Je kan een huidig proces/commando, hetzij in een bash terminal, hetzij in een powershell console, steeds stoppen met "ctrl + c".

Voorbereiding

We gaan er van uit dat je reeds een werkende VM voor Fedora en Windows Server hebt (zie vorige labo's).

Voor we de rest van dit labo kunnen uitvoeren moeten we eerst de scriptjes (te vinden op Chamilo, documenten => Praktijk => Processen) naar onze virtuele machine met Fedora en Windows Server kopiëren. Het bash script (extensie .sh) is voor Fedora bedoeld. De powershell scripts (extensie .ps1) zijn bedoeld voor de Windows Server VM.

Methode 1

Open in jouw virtuele machine een browser en ga naar de Chamilo cursus. Download via documenten de benodigde scriptjes afhankelijk van het OS van jouw VM en sla deze op in de map "Documents" van de virtuele machine.

Methode 2

Download de scriptjes in jouw HOST omgeving (= OS waarmee jouw computer opstart) en plaats ze op een USB-stick (FAT32). Koppel nadien de USB-stick in jouw GUEST omgeving (= VM) als volgt:

1. Selecteer "Devices" in de menubalk van jouw draaiende VM
2. Kies "USB" => <jouw USB device>
3. De USB-stick is nu gekoppeld binnen de draaiende VM en is toegankelijk via de bestandenverkenner in het OS van de VM
4. Kopieer de scriptjes, afhankelijk van het OS van de VM, van de USB-stick naar de "Documents" map

Processen in Linux

Overzicht commando's

Volgende tabel bevat een overzicht van alle commando's die aan bod zullen komen doorheen dit labo.

Omschrijving	Commando Linux (Bash)
Overzicht alle processen	ps
Overzicht processen per gebruiker	ps -aux
Overzicht details van een proces	ps <procesid>
Overzicht processtructuur	pstree
Overzicht alle actieve processen met automatische verversing	top
Id van proces zoeken	pidof <procesnaam>
Proces stopzetten	kill <pid>
Proces laten wachten	sleep <seconden>
Op achtergrond starten	<commando> &

Start je (virtuele) machine op met Fedora.

Na het aanmelden open je een nieuwe terminal. Dit kan op verschillende manieren:

- Toetsencombinatie: alt + F2 en voer dan "gnome-terminal" in en bevestig
- Grafisch: Klik linksboven op "Activiteiten", zoek naar terminal en klik om te openen

💡 Eens geopend kan je links in de lijst van favoriete applicaties rechtsklikken op je terminal en kiezen voor "Voeg toe aan favorieten". Zo kan je volgende keer snel vanuit deze lijst telkens een terminal starten.

Overzicht processen

We starten met het tonen van een overzicht van huidige processen die draaien op ons systeem:

```
ps
```

Als je het bovenstaande commando uitvoert zal je dus een lijst van processen te zien zijn, gelijkaardig met onderstaande afbeelding:

[sebastiaan@localhost ~]\$ ps			
PID	TTY	TIME	CMD
2149	pts/0	00:00:00	bash
2381	pts/0	00:00:00	ps

Via deze uitvoer kan je dus zien welke processen er actief zijn, maar enkel deze in de terminal zelf. Tevens zien we ook welk id elk proces heeft (PID). Daarnaast zien we ook welk commando gebruikt werd om het proces te starten en hoeveel CPU tijd het al heeft verbruikt.

Vervolgens zullen we het overzicht uitbreiden en tonen we meer informatie per proces, alsook de processen gebundeld per gebruiker. Hiervoor voeren we volgend commando uit:

```
ps -aux
```

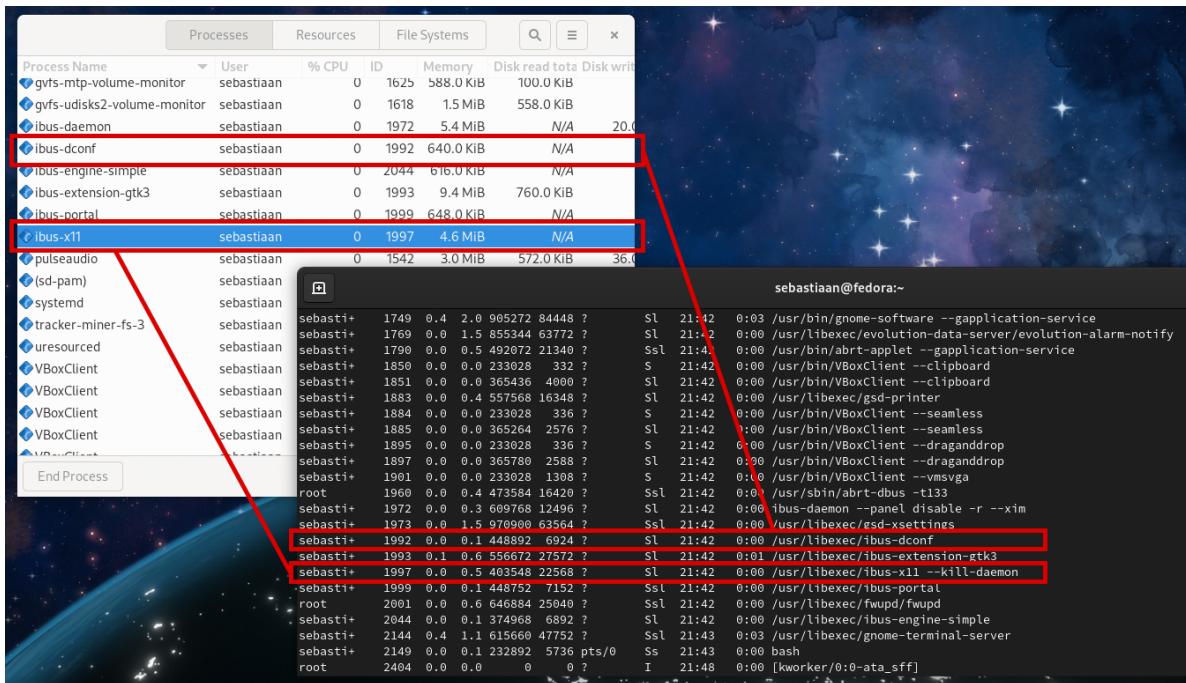
Dit levert jou onderstaand gelijkaardig overzicht op (echte uitvoer bevat veel meer processen):

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.4	109268	17444	?	Ss	21:42	0:01	/usr/lib/syst
root	2	0.0	0.0	0	0	?	S	21:42	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	21:42	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	21:42	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	21:42	0:00	[kworker/0:0H]
root	9	0.0	0.0	0	0	?	I<	21:42	0:00	[mm_percpu_wq]
root	10	0.0	0.0	0	0	?	S	21:42	0:00	[ksoftirqd/0]
root	11	0.0	0.0	0	0	?	I	21:42	0:00	[rcu_sched]
root	12	0.0	0.0	0	0	?	S	21:42	0:00	[migration/0]
root	13	0.0	0.0	0	0	?	S	21:42	0:00	[cpuhp/0]
root	14	0.0	0.0	0	0	?	S	21:42	0:00	[kdevtmpfs]
root	15	0.0	0.0	0	0	?	I<	21:42	0:00	[netns]
root	16	0.0	0.0	0	0	?	S	21:42	0:00	[rcu_tasks_kt]
root	17	0.0	0.0	0	0	?	S	21:42	0:00	[rcu_tasks_ru]
root	18	0.0	0.0	0	0	?	S	21:42	0:00	[rcu_tasks_tr]

Hieruit kunnen we heel wat meer informatie aflezen ten opzicht van het basiscommando `ps`

- User: toont de eigenaar van het proces
- %CPU en %MEM: procentuele belasting van de processor en het geheugen
- Start: tijdstip waarop het proces gestart werd

Je kan deze informatie ook vergelijken met de informatie die grafisch beschikbaar is via de "System Monitor". Open deze grafische applicatie (Activiteiten => Systeemmonitor). Zoek nu voor een aantal processen op basis van hun id de informatie op in zowel de terminal uitvoer als het grafische venster. Gevonden?



Probeer nu om de details van 1 van jouw eigen processen op te vragen in de shell. Welk commando gebruik je hiervoor?

Je kan ook via de terminal een overzicht opvragen van de meest belastende processen op jouw systeem. Dit doet je via het commando `top`

```
sebastiaan@fedora:~ — top
```

top - 12:11:33 up 47 min, 1 user, load average: 0.21, 0.20, 0.25
Tasks: 210 total, 1 running, 209 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.2 us, 1.8 sy, 0.0 ni, 92.5 id, 0.0 wa, 1.0 hi, 0.5 si, 0.0 st
MiB Mem : 3925.4 total, 1159.8 free, 1259.5 used, 1506.1 buff/cache
MiB Swap: 10154.0 total, 10154.0 free, 0.0 used. 2397.1 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
24485	sebasti+	20	0	4559488	472688	128788	S	8.3	11.8	3:26.12	gnome-shell
26239	sebasti+	20	0	618272	48960	38304	S	2.3	1.2	0:09.83	gnome-terminal
27159	sebasti+	20	0	597844	59200	43224	S	2.0	1.5	0:19.07	gnome-system-mo
62	root	20	0	0	0	0	I	0.3	0.0	0:00.71	kworker/u4:1-flush-8:0
27967	sebasti+	20	0	234992	5184	4272	R	0.3	0.1	0:00.10	top
1	root	20	0	110544	18364	11264	S	0.0	0.5	0:04.40	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
7	root	20	0	0	0	0	I	0.0	0.0	0:00.26	kworker/0:1-events
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_kthre
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude_
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
13	root	20	0	0	0	0	S	0.0	0.0	0:00.18	ksoftirqd/0
14	root	20	0	0	0	0	I	0.0	0.0	0:00.41	rcu_sched
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
18	root	rt	0	0	0	0	S	0.0	0.0	0:00.36	migration/1

Je kan het overzicht verlaten met de toets "q" of toetscombinatie "ctrl + c".

Processtructuur

Daarnaast kunnen we ook een overzicht van alle processen opvragen maar in boomstructuur. Het voordeel van deze structuur is dat we de onderlinge samenhang zien van processen. We zien met andere woorden de ouder van elk proces en zijn kinderen.

```
pstree
```

```
[sebastiaan@localhost ~]$ pstree
systemd--ModemManager---3*[{ModemManager}]
  NetworkManager---2*[{NetworkManager}]
  VBoxDRMClient
  VBoxService---8*[{VBoxService}]
  abrt-dbus---2*[{abrt-dbus}]
  3*[abrt-dump-journ]
  abrtd---2*[{abrtd}]
  accounts-daemon---3*[{accounts-daemon}]
  alsactl
  anacron
  atd
  auditd---{auditd}
  avahi-daemon---avahi-daemon
  chronyd
  colord---3*[{colord}]
  crond
  cupsd
  dbus-broker-lau---dbus-broker
  earlyoom
  firewalld---{firewalld}
  fwupd---4*[{fwupd}]
  gdm---gdm-session-wor---gdm-wayland-ses---gnome-session-b---3*[{gnome-session-b}]
    2*[{gdm-wayland-ses}]
    2*[{gdm-session-wor}]
    2*[{gdm}]
  gnome-keyring-d---3*[{gnome-keyring-d}]
  gssproxy---5*[{gssproxy}]
  mcelog
  packagekitd---2*[{packagekitd}]
  pcscd---6*[{pcscd}]
  polkitd---5*[{polkitd}]
  rngd---{rngd}
  rtkit-daemon---2*[{rtkit-daemon}]
  sssd---sssd_be
    sssd_nss
  sssd_kcm
  switcheroo-cont---2*[{switcheroo-cont}]
  systemd---(sd-pam)
    2*[VBoxClient---VBoxClient---2*[{VBoxClient}]]
    VBoxClient---VBoxClient---3*[{VBoxClient}]
    -VBoxClient
```

Wat valt nu op uit dit overzicht? De boomstructuur van de processen start vanaf "systemd". Dit is het hoofdproces van het besturingssysteem waar alle andere processen als kind aan gehangen worden. Deze informatie vinden we ook terug via `ps` omdat het rootproces "systemd" daar id 1 heeft. Ook de grafische omgeving (GDM) gebruikt nog zijn eigen kindprocessen om volledig te kunnen werken.

Nieuw proces opstarten

Start nu een aantal nieuwe grafisch processen op naar keuze, bijvoorbeeld: nieuwe terminal, bestanden, Firefox, Rekenmachine, ...) en probeer ze terug te vinden in de lijst van processen op de verschillende manieren zoals in vorige paragraaf besproken. Bespreek hieronder jouw werkwijze

Open niet alleen via de grafische manier, maar ook via de shell een nieuw proces "Firefox". Wat merk je op? Bekijk zeker de "system monitor" en uitvoer van "pstree".

Sluit alle openstaande programma's af, behalve de "terminal" & "system monitor". Start nu in jouw terminal een nieuw proces "gedit". Er wordt een grafische tekstverwerker (Gnome Editor) opgestart. Zoals je merkt is jouw terminal nu "bevroren". Omdat je het proces op de voorgrond gestart hebt zal jouw terminal niet meer reageren op nieuwe commando's zolang de tekstverwerker nog openstaat. Dit is uiteraard niet handig omdat we graag onze terminal nog altijd willen kunnen gebruiken. Dit kunnen we oplossen door het commando in de achtergrond van de terminal te starten. Sluit eerst "gedit" af en keer terug naar de terminal. Start "gedit" nu opnieuw op maar als achtergrond proces, als volgt:

```
gedit &
```

De tekstverwerker zal nog altijd gestart worden als voorheen maar we behouden nu de controle over de terminal en kunnen dus nieuwe instructies in de terminal ingeven en uitvoeren.

Als extra zullen we ook even kort toelichten hoe een proces automatisch gestart kan worden. We zullen in Linux inplannen dat elke minuut een nieuwe instantie, of nieuw document als er al een instantie open is, van "Gedit" geopend wordt. Dit zullen we doen door middel van [crontab](#). Standaard heb je geen toegang tot crontab dus zullen we eerst een nieuwe crontab aanmaken als volgt:

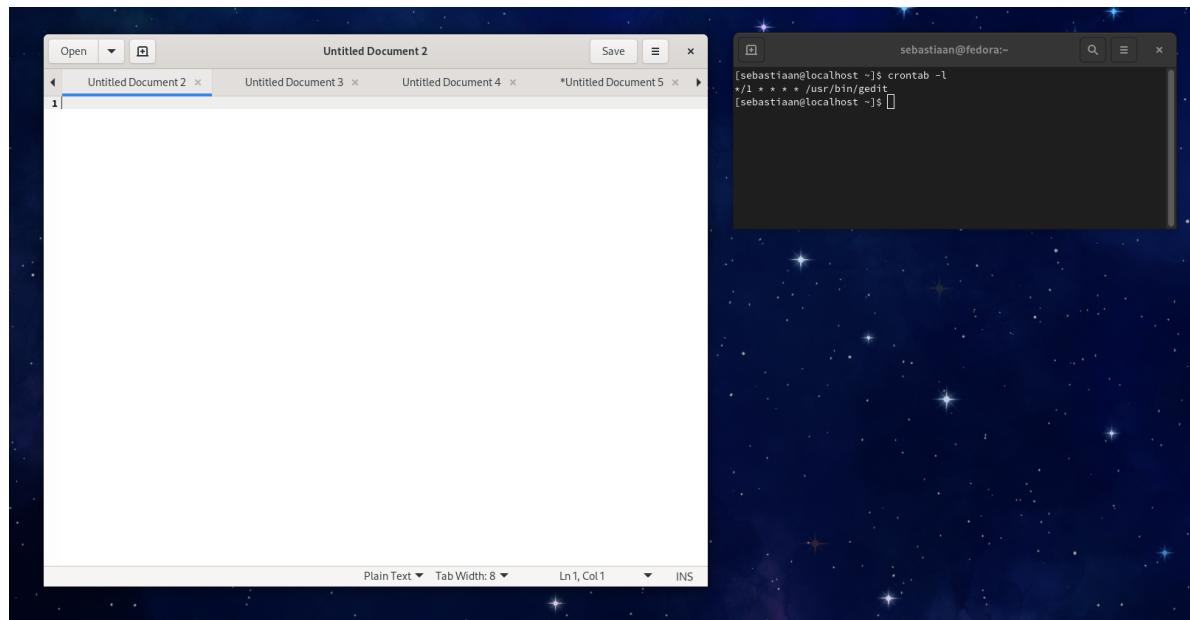
```
crontab -e
```

Een editor zou moeten openen met nog lege tekst. Plaats volgende tekst in het document `*/1 * * * * /usr/bin/gedit`. Sla de wijzigingen op met "ctrl + s" en sluit de editor af met "ctrl + x".

Controleer nadien in de terminal of de inhoud effectief goed werd doorgegeven aan crontab als volgt:

```
crontab -l
```

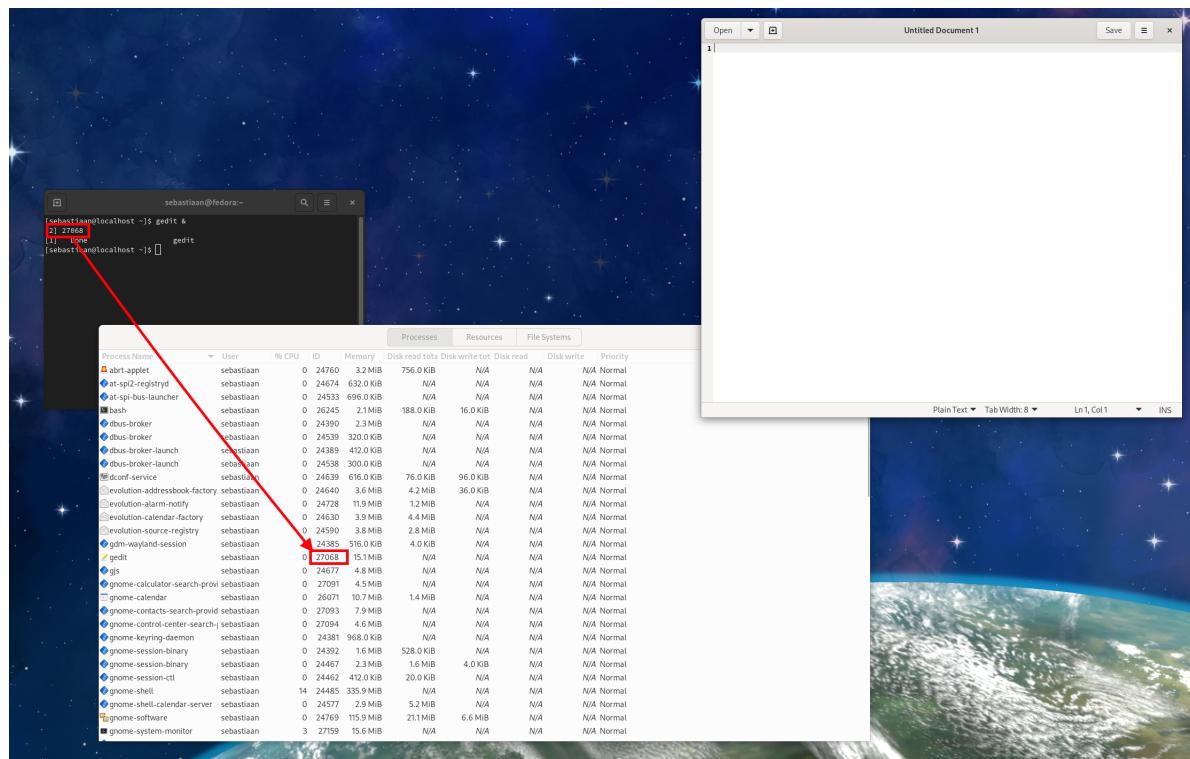
Nadien zal al alles goed gaan elke minuut een nieuwe "Gedit" geopend worden, of een nieuw document in "Gedit" als deze reeds geopend was.



Na een aantal minuten kan je het automatisch openen ongedaan maken door opnieuw `crontab -e` te gebruiken en de regel die je net toegevoegd hebt opnieuw weg te halen. Vergeet het document niet op te slaan na verwijderen van de regel en controleer of je crontab opnieuw leeg is!

Proces stoppen

Voorlopig hebben we enkel processen grafisch stopgezet door het "kruisje" rechtsboven bij de applicatie te gebruiken. In een serveromgeving zal je als beheerder nooit over een GUI beschikken maar moeten processen toch stopgezet kunnen worden. Hiervoor zullen het `kill` commando gebruiken. Open, indien je het reeds gesloten had, een nieuw proces "gedit" op de achtergrond via jouw terminal. Jouw terminal zal bij opstarten automatisch het procesid afdrukken. Dit id hebben we nodig om via het "kill" commando het proces te stoppen.



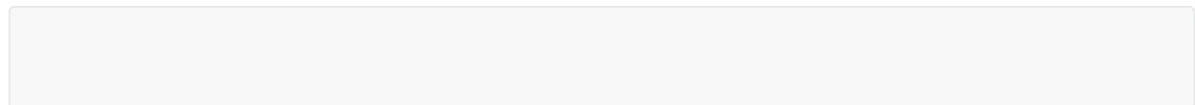
Voer nu in de terminal volgend commando uit waarbij je `je` vervangt door het procesid van jouw tekstverwerker.

```
kill <procesid>
```

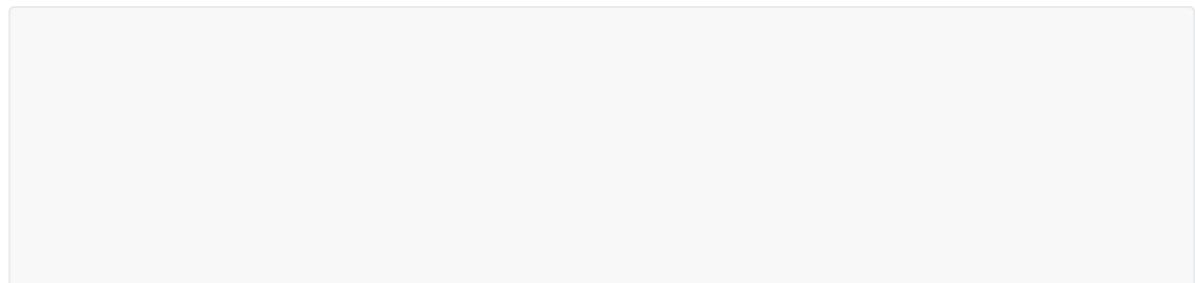
Als alles goed gaat zal "gedit" nu afgesloten zijn. Controleer dit via de commando's zoals reeds beschreven en via de "system monitor". Soms kan het zijn dat een proces toch niet zomaar afgesloten kan worden. Je kan dit dan alsnog "forceren" door een extra vlag mee te geven aan het `kill` commando als volgt:

```
kill -9 <procesid>
```

Probeer bovenstaand commando uit om jouw huidig geopende terminal af te sluiten.



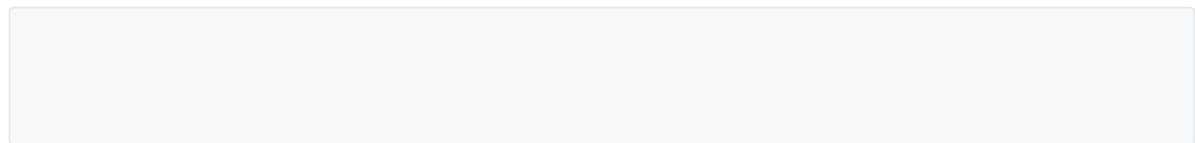
Open nu een nieuwe terminal en ga op zoek naar het procesid van het hoofdproces van jouw huidige sessie. Probeer deze af te sluiten. Wat is het effect? Wat zou de reden kunnen zijn?



Als variant op het stopzetten van een proces kan je het ook pauzeren. Dit doe je als volgt

```
sleep <aantal seconden>
```

Probeer nu jouw terminal proces 5 seconden te laten slapen. Je zal merken dat de terminal opnieuw bevroren is. Nadien probeer je de terminal op de achtergrond te laten slapen. Hoe doe je dit beide?



Zorg er nu voor dat je in de terminal in de map "Documents" zit, want we zullen nu het script uitvoeren dat je gedownload hebt tijdens de voorbereiding naar deze map. Om zeker te zijn dat je in de juiste map zit voer je volgende commando's uit:

```
cd  
cd ./Documents
```

Nadien voer je onderstaand commando uit om het scriptje te starten:

```
bash StartLoop.sh
```

Jouw terminal zal in een oneindige lus blijven hangen en elke seconde tekst produceren. Sluit het commando af op een manier naar keuze zonder de terminal af te sluiten. Als laatste sluit je jouw VM af door in de terminal het commando `poweroff` in te voeren. Dit zal alle processen afsluiten en jouw virtuele machine stopzetten.

Processen in Windows

Overzicht commando's

Volgende tabel bevat een overzicht van alle commando's die aan bod zullen komen doorheen dit labo.

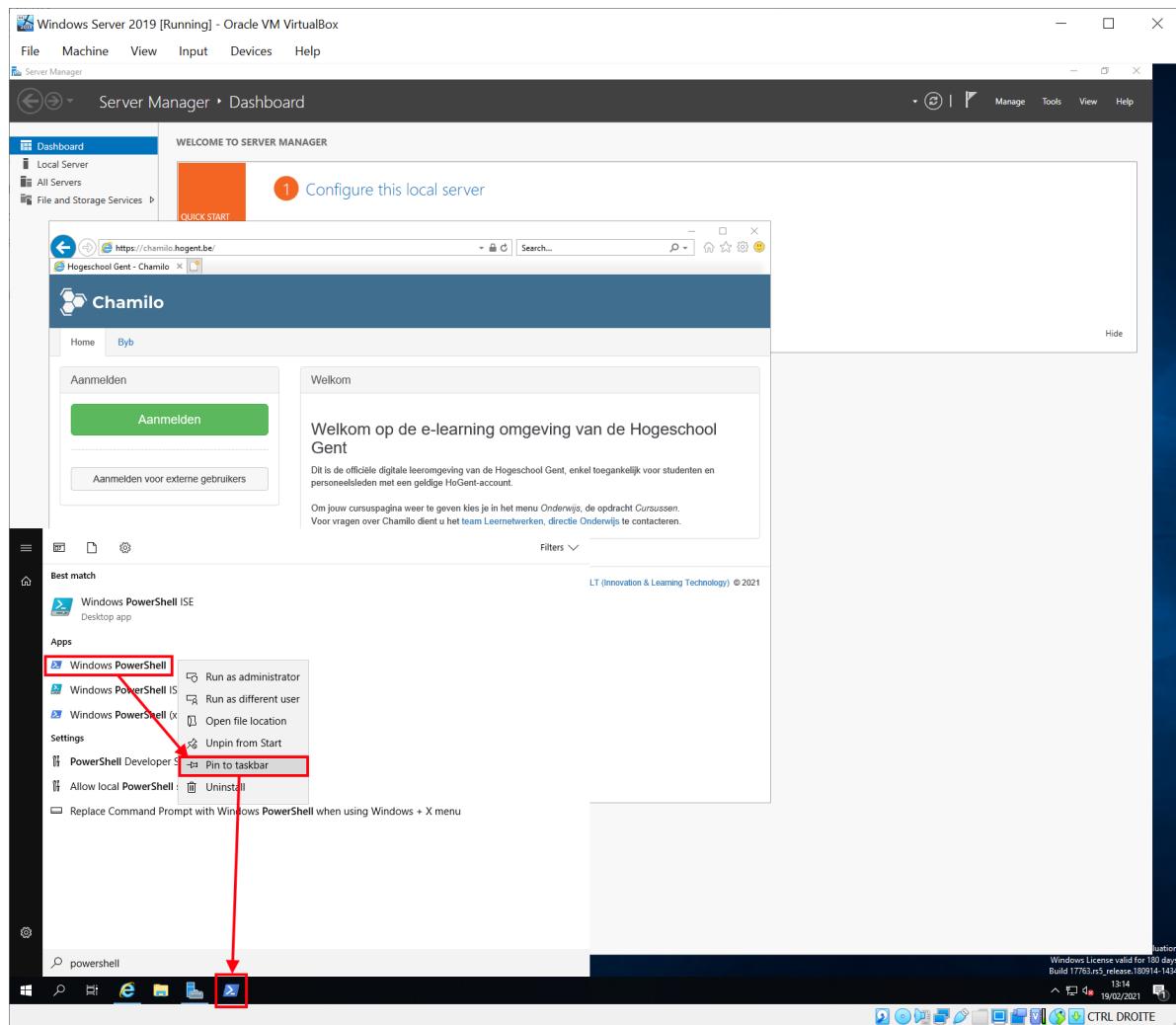
Omschrijving	Commando Windows (Powershell)
Overzicht alle processen	Get-Process
Overzicht processen per gebruiker	Get-UserProcess.ps1 (*)
Overzicht details van een proces	Get-Process -Id <procesid>
Overzicht processtructuur	Get-ProcessTree.ps1 (*)
Overzicht alle actieve processen met automatische verversing	Get-ProcessTop.ps1 (*)
Id van proces zoeken	(Get-Process -Name <procesnaam>).Id
Proces stopzetten	Stop-Process <id>
Proces laten wachten	Start-Sleep <seconden>
Proces op achtergrond starten	Start-Process -NoNewWindow <commando> Start-Job { <commando> }

Commando's aangeduid met (*) zijn geen standaard ingebouwde commando's in Powershell. Dit zijn zelfgemaakte scriptjes die je kan terugvinden op Chamilo. Om deze uit te voeren plaats je de bestanden in een map op jouw virtuele machine. Vervolgens ga je in PowerShell naar die map (gebruik commando `cd`) en voer je het script uit als volgt: `.\<bestandsnaam>`. Bijvoorbeeld: om een overzicht van processen van de gebruiker te krijgen voer je in de Powershell console het volgende commando uit: `.\Get-UserProcess.ps1`. Er van uitgaande dat het bestand "Get-Processes.ps1" in de huidige map staat!

Powershell terminal pinnen

Nu zullen we net zoals in Linux de terminal (Powershell) pinnen aan de taakbalk zodat we deze later sneller kunnen openen. Dit doen we als volgt:

1. Klik op het Windows logo links onderaan
2. Typ "Powershell"
3. Klik rechts op "Windows Powershell" en kies voor "Pin to taskbar"
4. Je hebt nu onderaan in jouw taakbalk het "Windows Powershell" icoontje



Powershell toestaan externe scripts uit te voeren

Aangezien we als "administrator" inloggen op onze server zullen we altijd de mogelijkheid hebben om ook scriptjes die niet eigen zijn aan Windows uit te voeren. Op jouw eigen toestel zal het echter niet altijd lukken om zomaar andere scriptjes uit te voeren. De "ExecutionPolicy" moet immers minstens de waarde "RemoteSigned" hebben. Controleer deze vlag op jouw Windows Server VM door volgend commando in te geven in Powershell:

```
Get-ExecutionPolicy
```

Op jouw server zou deze standaard de waarde "RemoteSigned" moeten teruggeven. Indien dit niet het geval is kan je die waarde instellen via volgend commando en kies voor "A" als antwoord:

```
Set-ExecutionPolicy "RemoteSigned"
```

Controleer nadien of deze waarde effectief werd ingesteld met eerder genoemd commando. Je voert deze stappen ook uit op jouw HOST toestel (indien Windows als OS) zodat je ook de oefeningen in latere labo's of voor andere vakken in Powershell kan uitvoeren. Indien de scriptjes toch nog niet zouden uitvoeren gebruik je als "ExecutionPolicy" best "unrestricted". Deze setting gebruik je best **enkel in de VM** en dus niet op jouw HOST toestel.

Overzicht processen

Zorg er voor dat je in Powershell in dezelfde map zit als waar je de scriptjes hebt opgeslagen, dit zou "Documents" moeten zijn. **Blijf in deze map voor de rest van het labo!** Voer volgend commando uit om naar de map te gaan:

```
cd c:\Users\Administrator\Documents
```

We starten met het tonen van een overzicht van huidige processen die draaien op ons systeem. Voer hiervoor volgend commando uit:

```
.\Get-UserProcess.ps1
```

Als je het bovenstaande commando uitvoert zal een lijst van processen te zien zijn, gelijkaardig met onderstaande afbeelding:

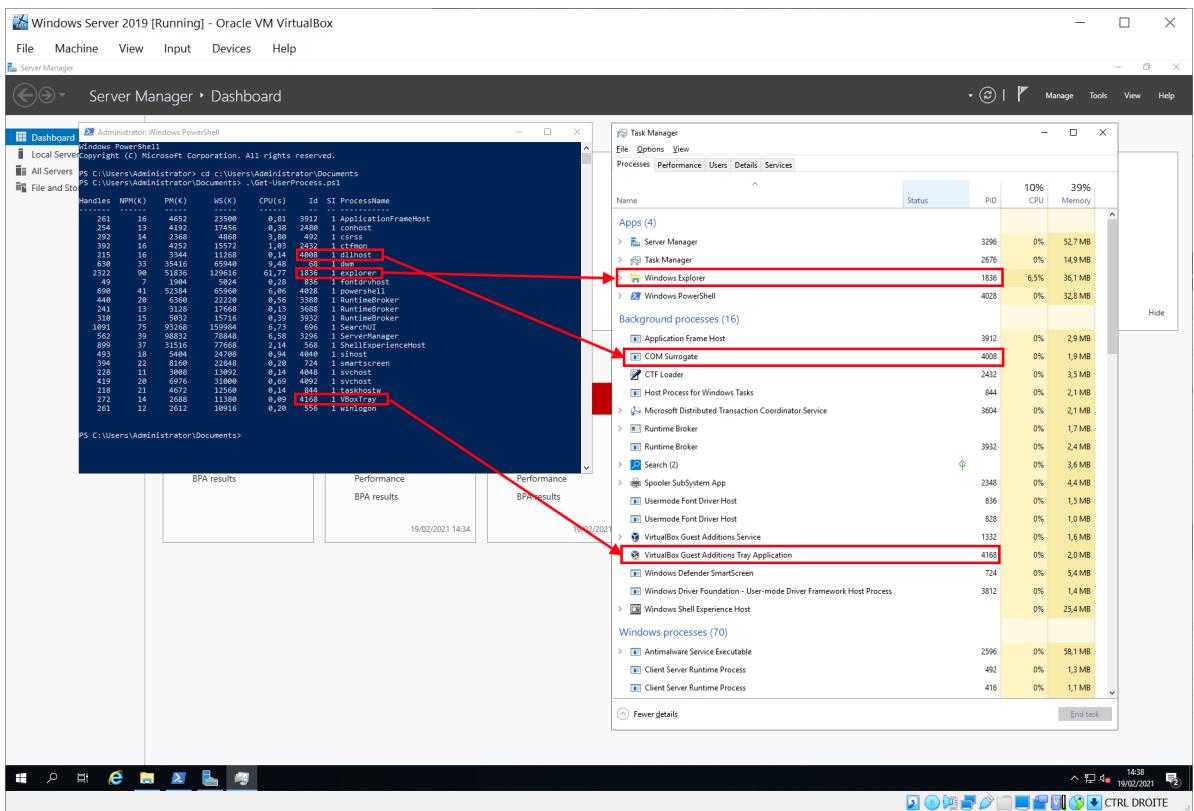
The screenshot shows a Windows Server 2019 environment within Oracle VM VirtualBox. The PowerShell window title is 'Administrator: Windows PowerShell'. The command run was '.\Get-UserProcess.ps1'. The output is a table of processes:

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
261	16	4652	23500	0,81	3912	1	ApplicationFrameHost
254	13	4192	17456	0,38	2480	1	conhost
292	14	2368	4868	3,80	492	1	csrss
392	16	4252	15572	1,03	2432	1	ctfmon
215	16	3344	11268	0,14	4008	1	dllhost
630	33	35416	65940	9,48	68	1	dwm
2322	90	51836	129616	61,77	1836	1	explorer
49	7	1904	5024	0,28	836	1	fontdrvhost
690	41	52384	65960	6,06	4928	1	powershell
440	20	6360	22220	0,56	3388	1	RuntimeBroker
241	13	3128	17668	0,13	3680	1	RuntimeBroker
310	15	5032	15716	0,39	3932	1	RuntimeBroker
1091	75	93268	159984	6,73	696	1	SearchUI
562	39	98832	78848	6,58	3296	1	ServerManager
899	37	31516	77668	2,14	568	1	ShellExperienceHost
493	18	5404	24708	0,94	4940	1	sihost
394	22	8168	22848	0,20	724	1	smartscreen
228	11	3008	13092	0,14	4948	1	svchost
419	20	6976	31000	0,69	4992	1	svchost
218	21	4672	12560	0,14	844	1	taskhostw
272	14	2688	11380	0,09	4168	1	VboxTray
261	12	2612	10916	0,20	556	1	winlogon

Let op: niet alle processen worden via deze weg getoond!

Via deze uitvoer zien we als belangrijkste gegevens het id dat elk proces heeft (Id), de verbruikte CPU tijd (CPUs) en de naam van het proces (ProcessName)

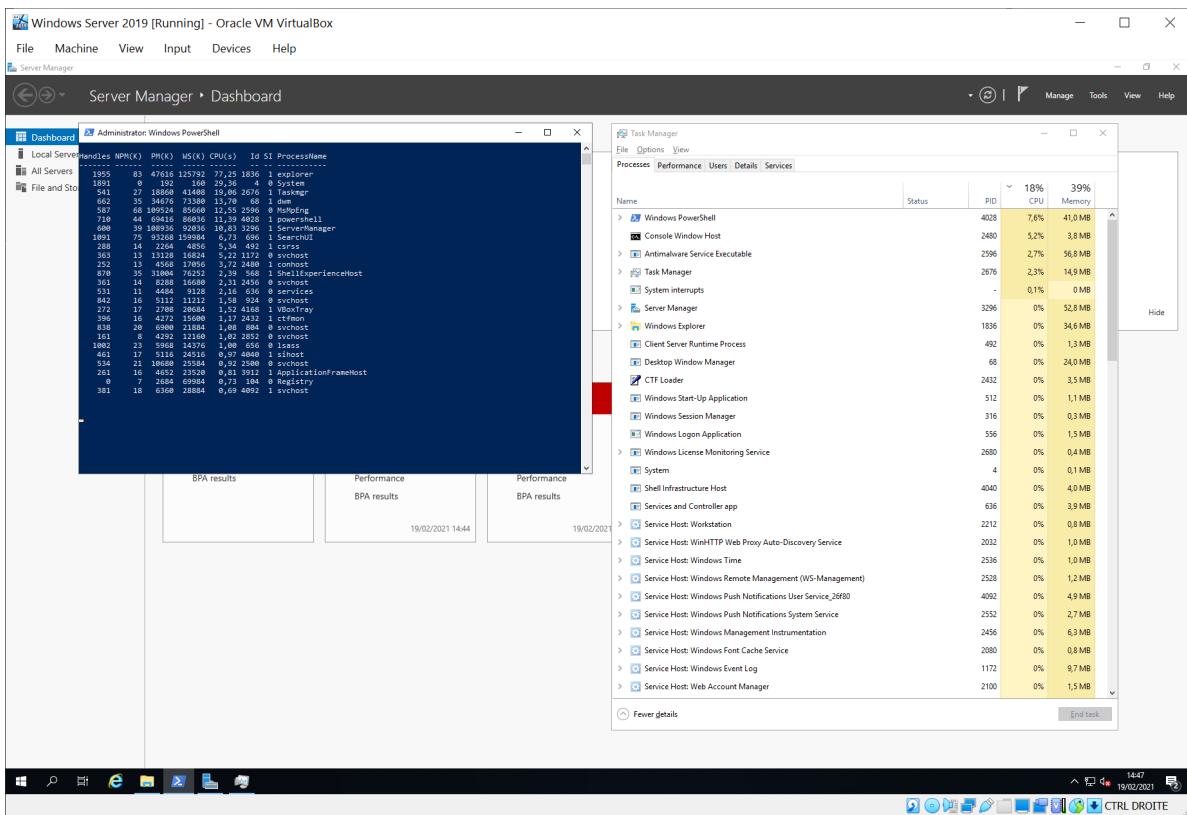
Je kan ook deze informatie vergelijken met de informatie die grafisch beschikbaar is via de "Task Manager". Open deze grafische applicatie (Rechts klikken op taakbalk => Task Manager) en open "More Details". Activeer nu de kolom voor "PID" door rechts te klikken op een kolomnaam en deze aan te vinken. Zoek nu voor een aantal processen op basis van hun id de informatie op in zowel de terminal uitvoer als het grafische venster. Gevonden? Zoals je merkt komt de naam niet altijd overeen!



Probeer nu om de details van 1 van jouw eigen processen op te vragen in de shell. Wat is hiervoor het commando?

```
.\Get-ProcessTop.ps1
```

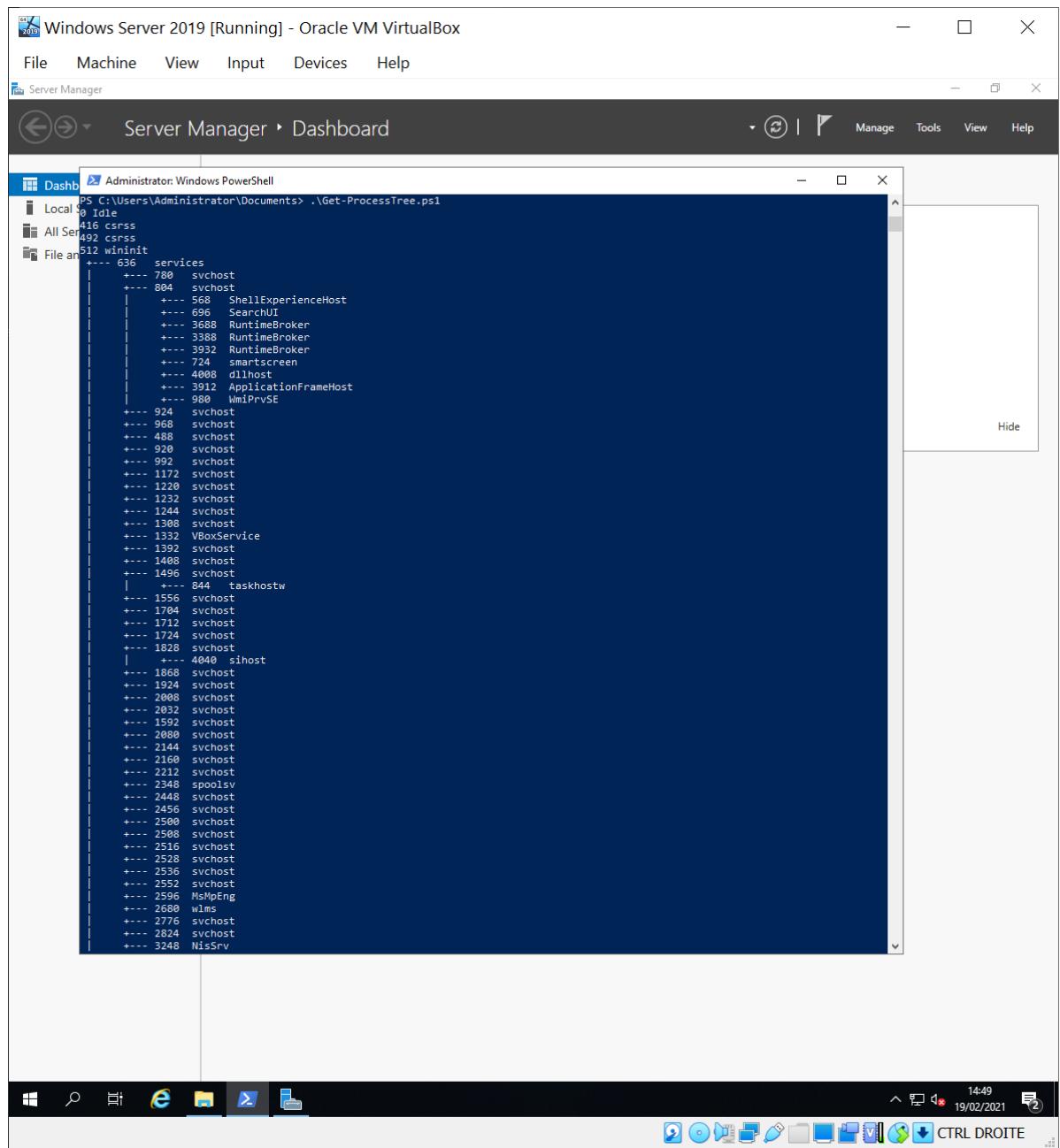
Je kan ook via de terminal een overzicht vragen van de meest belastende processen op jouw systeem. Dit doe je door het script "Get-ProcessTop" uit voeren. De uitvoer wordt elke seconde automatisch bijgewerkt. Stoppen doe je via de toetscombinatie "ctrl + c".



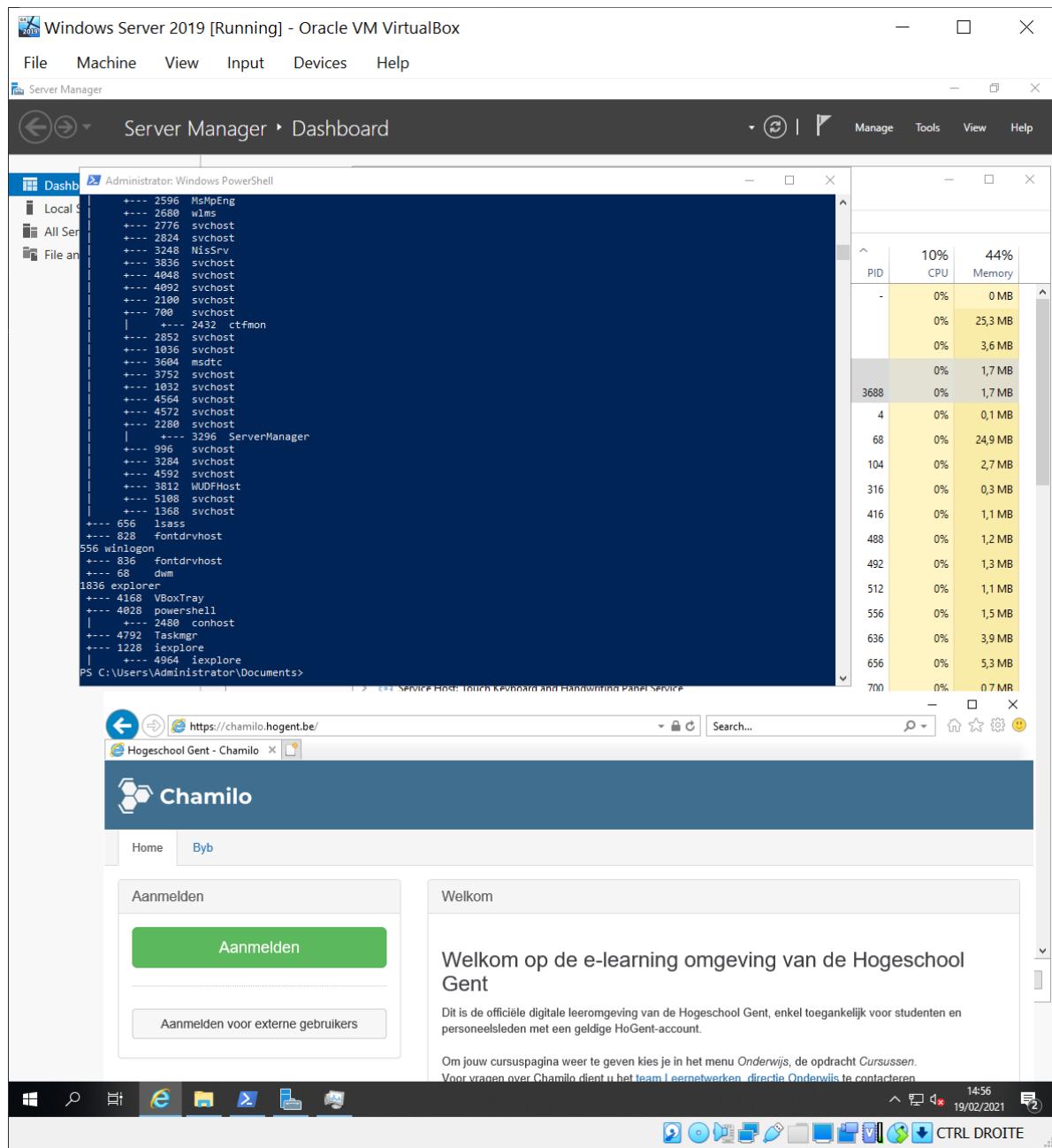
Processtructuur

We kunnen daarnaast ook een overzicht krijgen van alle processen in een boomstructuur. Het voordeel van deze structuur is dat we de onderlinge samenhang zien van processen. We zien met andere woorden de ouder van elk proces en zijn kinderen. Hiervoor voeren we het script "Get-ProcessTree" uit.

```
.\Get-ProcessTree.ps1
```



Wat valt op in dit overzicht? De boomstructuur van de processen heeft een aantal hoofdtakken die starten vanaf "wininit", "winlogon" en "explorer". Deze laatste tak bevat alle processen die jij als gebruiker hebt gestart of dit automatisch voor jou werden gestart bij aanmelden (bv.: vboxtray). Open bijvoorbeeld "Internet Explorer" en voer daarna het script voor de processtructuur opnieuw uit, je zal zien dat "iexplore" werd toegevoegd onder de hoofdtak "explorer".

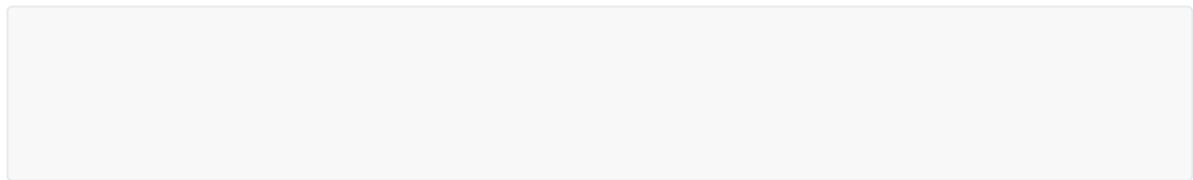


Nieuw proces opstarten

Start nu een aantal nieuwe grafische processen op naar keuze, bijvoorbeeld: nieuwe powershell terminal, verkenner, browser, Rekenmachine, ...) en probeer ze terug te vinden in de lijst van processen op de verschillende manieren zoals in vorige paragraaf besproken. Bespreek hieronder jouw werkwijze

Je kan ook een proces (of toepassing) rechtstreeks starten vanuit de powershell terminal. Probeer op deze manier een nieuwe kladblok en rekenmachine te starten.

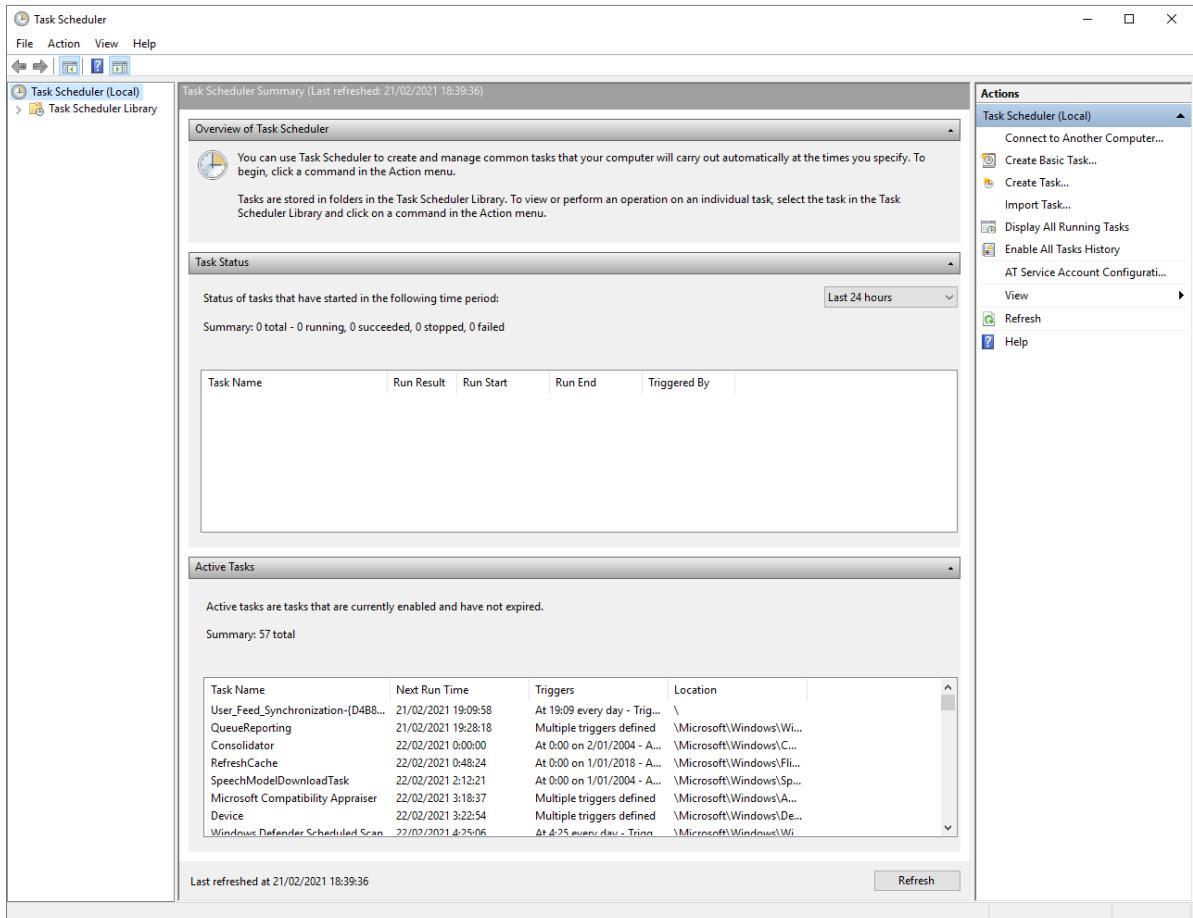
💡 Om het commando te achterhalen kan je het proces eerst grafisch starten en dan via de "Task Manager" proberen de naam te achterhalen.



Net zoals Linux kunnen we ook in Windows processen op de achtergrond starten in de Powershell terminal.

```
Start-Process -NoNewWindow <commando>
```

Als extra zullen we ook even kort toelichten hoe een proces automatisch gestart kan worden. We zullen in Windows inplannen dat elke minuut een nieuwe rekenmachine geopend wordt. Open hiervoor de "Task Scheduler".

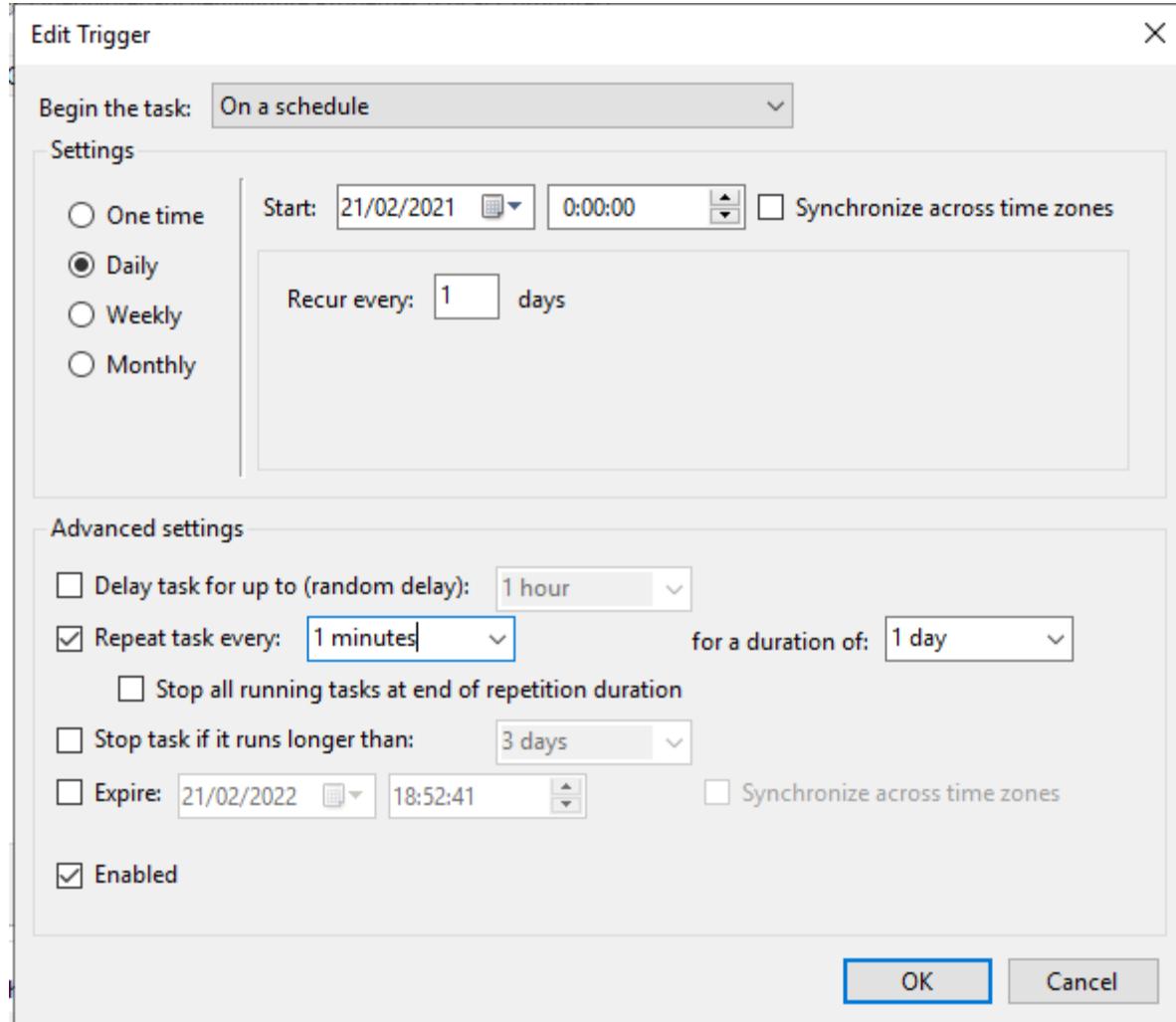


Je krijgt hier een overzicht van alle actieve taken die reeds gepland zijn en indien er taken gelopen hebben kan je hier ook de details van raadplegen.

Om elke minuut een instantie van de rekenmachine te starten zullen we dus een nieuwe taak moeten toevoegen:

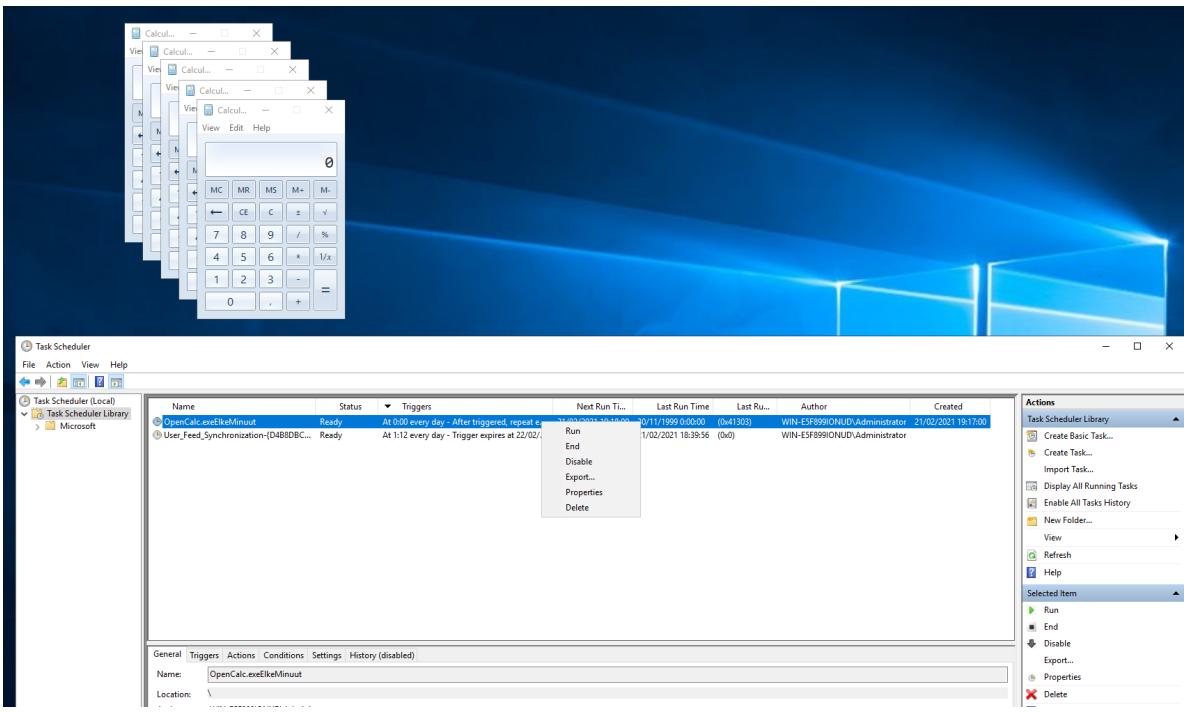
1. Klik op "Create Basic Task"
2. Create a basic task: Kies zelf een naam naar keuze en vul deze in. Kies dan "next"
3. Trigger: selecteer "Daily" en kies dan "next"
4. Daily: verander de tijd naar "00:00:00" en kies dan "next"
5. Action: "Start a program" en ga verder

6. Start a program: vul "calc.exe" in onder "Program/script" en kies dan "next"
7. Finish: vink het vakje aan voor "Open the properties dialog for this task when I click finish" en kies "Finish"
8. In het eigenschappenvenster van de taak ga je naar het tabblad "triggers"
9. Selecteer jouw trigger en kies "Edit"
10. Pas nu de velden aan zoals in onderstaande afbeelding en kies dan "Ok"



11. Keer in het vorige scherm terug naar het tabblad "Settings"
12. Vink hier "Run task as soon as possible after scheduled start is missed"
13. Klik tenslotte op "Ok" om het eigenschappenvenster af te sluiten

Nu zou er elke minuut een instantie van "calc.exe" moeten geopend worden. Na een aantal minuten kan je het automatisch openen ongedaan maken door in het scherm van "Task Scheduler" in de boomstructuur links voor "Task Scheduler Library" te kiezen. Rechts in het overzicht kies je dan jouw aangemaakte taak. Klik rechts op deze taak en kies voor "delete".

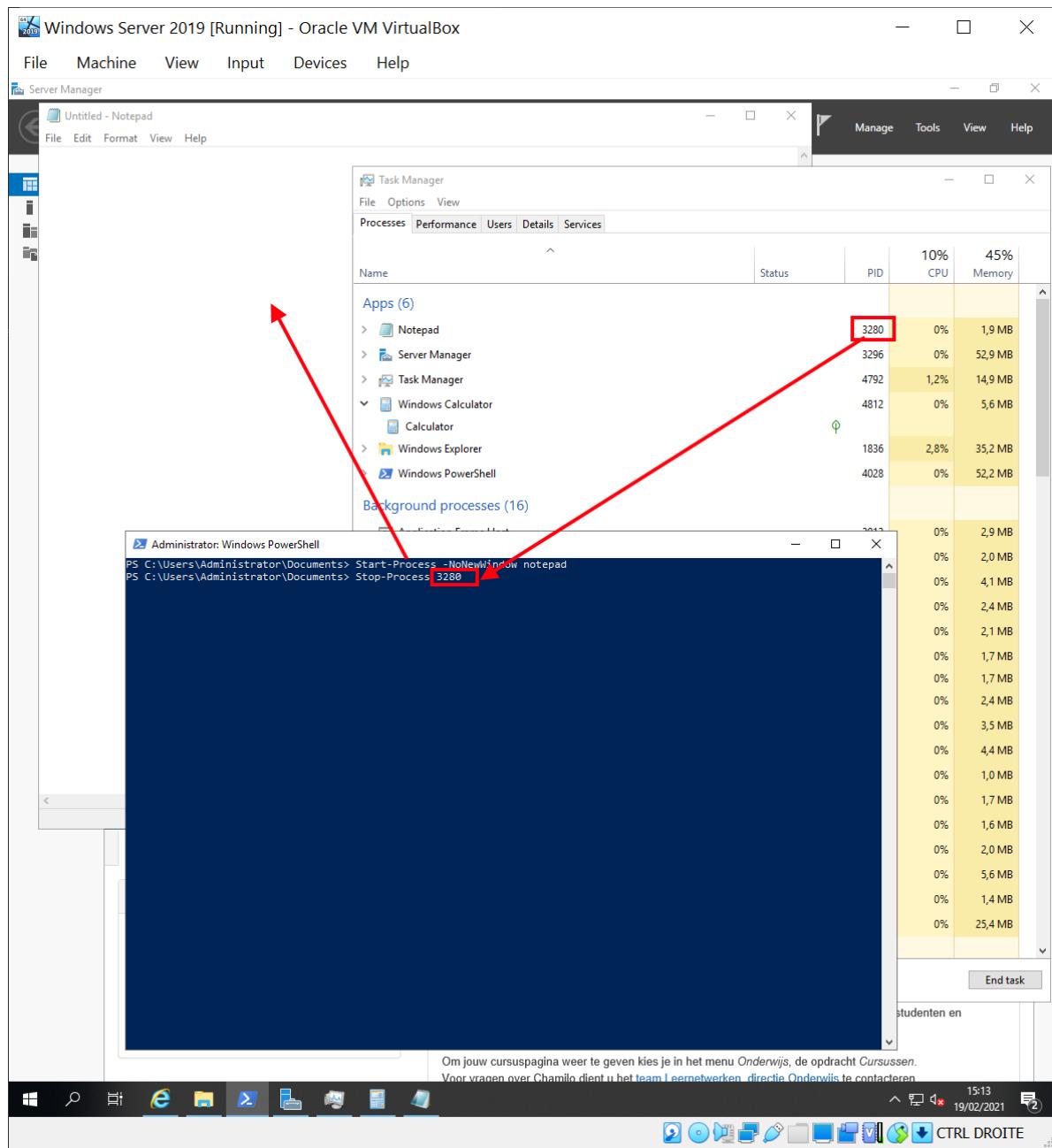


Proces stoppen

Voorlopig hebben we enkel processen grafisch stopgezet door het "kruisje" rechtsboven bij de applicatie te gebruiken. Het is ook mogelijk om via de "Task Manager" een proces stop te zetten. Dit doe je door rechts te klikken op de naam van een proces en dan te kiezen voor "End Task". In een serveromgeving zal je als beheerder nooit over een GUI beschikken maar moeten processen toch stop kunnen gezet worden. Hiervoor zullen het commando `stop-Process <proces_id>` gebruiken. Open, indien je deze reeds gesloten had, een nieuw proces "notepad" op de achtergrond via jouw terminal. Je krijgt, in tegenstelling tot Linux, hier niet onmiddellijk het id van het gestarte proces te zien. Zoek dit dus zelf op!

Voer nu in de terminal volgend commando uit waarbij je `<procesid>` vervangt door het procesid van jouw tekstverwerker.

```
Stop-Process <procesid>
```



Als alles goed gaat zal "notepad" nu afgesloten zijn. Controleer dit via de commando's zoals reeds beschreven en via de "Task Manager".

Als variant op het stopzetten van een proces kan je het ook pauzeren. Dit doe je als volgt:

```
Start-Sleep <aantal seconden>
```

Probeer nu jouw terminal proces 5 seconden te laten slapen. Je zal merken dat jouw terminal opnieuw bevroren is. Nadien probeer je jouw terminal op de achtergrond te laten slapen. Hoe doe je dit?

Voer nu als toemaatje volgend commando uit:

```
.\StartLoop.ps1
```

Jouw terminal zal in een oneindige lus blijven hangen en elke seconde tekst produceren. Sluit het commando af op een manier naar keuze zonder de terminal af te sluiten. Als laatste sluit je jouw VM af door in de terminal het commando `stop-computer` in te voeren. Dit zal alle processen afsluiten en jouw virtuele machine stopzetten.