



System Management

Deel I : scheduling

**DE HOGESCHOOL
MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Tijd en tijdsnotatie corrigeren

- Tijd corrigeren
 - indien de tijdzone niet correct is
(juist 1 of 2 uur verschil)
 - `sudo timedatectl set-timezone Europe/Brussels`
- Tijdsnotatie corrigeren
 - indien de dag bvb in het duits wordt
meegegeven
 - `sudo locale-gen nl_BE.UTF-8`
 - `sudo sed -i 's/de_BE/nl_BE/' /etc/default/locale`
 - de_BE kan natuurlijk iets anders zijn



one time jobs with at

at : een commando laten uitvoeren op een bepaald tijdstip

(handig bij zware programma's, die best 's nachts worden uitgevoerd)

```
student@ubdesk:~$ date
wo dec 18 14:16:56 CET 2024
student@ubdesk:~$ at 14:20
warning: commands will be executed using /bin/sh
at> echo "It is now $(date +%T) on $(date +%A)" > testat
at> <EOT>
job 1 at Wed Dec 18 14:20:00 2024
student@ubdesk:~$ ls test*
ls: cannot access 'test': No such file or directory
student@ubdesk:~$ date
wo dec 18 14:20:21 CET 2024
student@ubdesk:~$ ls test*
testat
student@ubdesk:~$ cat testat
It is now 14:20:00 on woensdag
student@ubdesk:~$
```

Ctrl-D → <EOT>

→ warning van /bin/sh is niet helemaal waar.

→ Indien in script de shebang /bin/bash staat, wordt het in bash uitgevoerd!



one time jobs with at

at verstaat (beperkt) Engels

```
student@ubdesk:~$ at 20:00 next week
warning: commands will be executed using /bin/sh
at> sleep 5
at> <EOT>
job 3 at Wed Dec 25 20:00:00 2024
```

atq of at -l

```
student@ubdesk:~$ atq
3    Wed Dec 25 20:00:00 2024 a student
student@ubdesk:~$ at -l
3    Wed Dec 25 20:00:00 2024 a student
```



one time jobs with at

atrm

om een job uit de queue te verwijderen

```
student@ubdesk:~$ at teatime
warning: commands will be executed using /bin/sh
at> sleep 5
at> <EOT>
job 4 at Wed Dec 18 16:00:00 2024
student@ubdesk:~$ atq
3    Wed Dec 25 20:00:00 2024 a student
4    Wed Dec 18 16:00:00 2024 a student
student@ubdesk:~$ atrm 4
student@ubdesk:~$ atq
3    Wed Dec 25 20:00:00 2024 a student
```



one time job with at

- `/etc/at.allow`
 - lijst van users die jobs mogen plannen met at
 - wie niet in deze file staat, mag geen jobs plannen
 - een lege allow-file wil zeggen dat niemand mag plannen
 - als deze file niet bestaat wordt de `at.deny` gechecked
- `/etc/at.deny` (bestaat standaard met heel wat system-users)
 - lijst van users die geen jobs mogen uitvoeren met at
 - wie niet in deze file staat, mag jobs plannen
 - een lege deny-file wil zeggen dat iedereen mag plannen



Als beide files niet bestaan kan iedereen at gebruiken

cron

- crontab file

- crontab commando wordt gebruikt om deze file te onderhouden
- elke user heeft zijn eigen crontab file (met als pad: `/var/spool/cron/crontabs/<username>`)
- jobs op specifieke tijdstippen herhalen
- 5 velden om dit tijdstip te bepalen

minute hour day of the month month day of the week

* => alle waarden voor dit veld

alternatief voor deze 5 velden:

@reboot, @yearly, @annually, @monthly, @daily, @midnight of @hourly



cron

voorbeeld crontab file

commando elke dag om 20:30 uitvoeren

```
30 20 * * * echo $(date) >> /tmp/testcron
```

elke 1e dag van de maand om 20:30

```
30 20 1 * * echo $(date) >> /tmp/testcron
```

elke zondag om 20:30

```
30 20 * * 7 echo $(date) >> /tmp/testcron
```



→ *day of the week: zondag -> zowel 0 als 7 werkt*

cron

voorbeeld crontab file

op maandag tem woensdag en zaterdag en zondag om 20:30

```
30 20 * * 1-3,6-7 echo $(date) >> /tmp/testcron
```

elke 5e minuut

```
*/5 * * * * echo $(date) >> /tmp/testcron
```

om de 10 minuten vanaf de derde minuut

```
3,13,23,33,43,53 * * * * echo $(date) >> /tmp/testcron
```



cron

voorbeeld crontab file

<u>String</u>	<u>Meaning</u>
@reboot	Run once, at startup.
@yearly	Run once a year, "0 0 1 1 *".
@annually	(same as @yearly)
@monthly	Run once a month, "0 0 1 * *".
@weekly	Run once a week, "0 0 * * 0".
@daily	Run once a day, "0 0 * * *".
@midnight	(same as @daily)
@hourly	Run once an hour, "0 * * * *".



voorbeeld van @reboot in crontab-file

```
@reboot echo "reboot at $(date)" >> /home/student/testreboot
```

cron

crontab commando

crontab file niet rechtstreeks editeren => crontab -e

```
student@ubdesk:~$ crontab -e
no crontab for student - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 2
```



crontab file tonen => crontab -l

cron

- cron.allow en cron.deny
 - in /etc
 - analoog met at.allow en at.deny



cron

/etc/crontab

dit is de **algemene** crontab-file (niet per user)

```
student@ubdesk:~$ cat /etc/crontab
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
```

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
# m h dom mon dow user  command
```

```
17 * * * *      root    cd / && run-parts --report /etc/cron.hourly
```

```
25 6 * * *      root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
```

```
47 6 * * 7      root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
```

```
52 6 1 * *      root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

```
#
```

```
student@ubdesk:~$
```

cron

cron files en directories

```
student@ubdesk:~$ ls /etc/cron.*
```

```
/etc/cron.d:
```

```
anacron  popularity-contest
```

```
/etc/cron.daily:
```

```
0anacron  apt-compat  cracklib-runtime  google-chrome  man-db  passwd  ubuntu-advantage-tools  
apport    bsdmainutils  dpkg  logrotate      mlocate  popularity-contest  update-notifier-common
```

```
/etc/cron.hourly:
```

```
/etc/cron.monthly:
```

```
0anacron
```

```
/etc/cron.weekly:
```

```
0anacron  man-db  update-notifier-common
```



cron.daily → dagelijks om 6:25 (zie vorige slide)

anacron

period	delay	job-identifier	command
--------	-------	----------------	---------

period	→ 1=@daily	7=@weekly	30=@monthly
delay	→ aantal minuten vooraleer job te starten		
jobid	→ naam weggeschreven in de timestamp-files		
command	→ uit te voeren commando		

/etc/anacrontab

indien dit bestaat worden verschillende regels van de algemene crontab niet uitgevoerd

anacron wordt uitgevoerd telkens het OS opstart

```
student@ubdesk:~$ cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
HOME=/root
LOGNAME=root

# These replace cron's entries
1    5    cron.daily    run-parts --report /etc/cron.daily
7    10   cron.weekly  run-parts --report /etc/cron.weekly
@monthly 15   cron.monthly  run-parts --report /etc/cron.monthly

student@ubdesk:~$
```

anacron

/var/spool/anacrontab/*

```
student@ubdesk:~$ date
Thu Oct 17 15:27:23 CEST 2024

student@ubdesk:~$ ls /var/spool/anacron/*
/var/spool/anacron/cron.daily
/var/spool/anacron/cron.weekly
/var/spool/anacron/cron.monthly

student@ubdesk:~$ sudo cat /var/spool/anacron/cron.daily
20241017

student@ubdesk:~$ sudo cat /var/spool/anacron/cron.weekly
20241014

student@ubdesk:~$ sudo cat /var/spool/anacron/cron.monthly
20241007

student@ubdesk:~$
```

Anacron controleert of een job is uitgevoerd binnen de periode gespecificeerd in het period-veld

- Indien niet voert hij het commando uit na een wachttijd gespecificeerd in het delay-veld
- Eens een job is uitgevoerd plaatst hij een timestamp in de file die gespecificeerd is in het jobid-veld

anacron

/etc/anacrontab

we voegen een job toe

period	delay	job-identifier	command
--------	-------	----------------	---------

period	→ 1=@daily 7=@weekly 30=@monthly
delay	→ aantal minuten vooraleer job te starten
jobid	→ naam weggeschreven in de timestamp-files
command	→ uit te voeren commando

```
student@ubdesk:~$ cat /etc/anacrontab
```

```
...
```

```
# These replace cron's entries
```

```
1    5    cron.daily    run-parts --report /etc/cron.daily
7    10   cron.weekly   run-parts --report /etc/cron.weekly
@monthly 15   cron.monthly  run-parts --report /etc/cron.monthly
1    5    my.backup     tar -czf /tmp/hfs_backup.tar.gz    /home
```

```
student@ubdesk:~$ ls /var/spool/anacron/
cron.daily    cron.weekly    cron.monthly
```

```
student@ubdesk:~$ anacron
```

```
student@ubdesk:~$ ls /var/spool/anacron/
cron.daily    cron.weekly    cron.monthly    my.backup
```

```
student@ubdesk:~$ ls /tmp/*.tar/gz          #pas na 5 minuten
/tmp/hfs_backup.tar.gz
```



System Management

Deel II : memory management

**DE HOGESCHOOL
MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



memory management

memory en cache tonen

```
student@ubdesk:~$ head -5 /proc/meminfo
MemTotal:      4015640 kB
MemFree:       805728 kB
MemAvailable:  2445544 kB
Buffers:       124044 kB
Cached:        1653624 kB
```

MemTotal: totale hoeveelheid fysieke RAM

MemFree: ongebruikt RAM

MemAvailable: hoeveelheid RAM dat kan gebruikt worden om programma's op te starten zonder swapruimte te moeten gebruiken

Buffers: RAM gebruikt voor buffering files

Cached: hoeveelheid RAM gebruikt als cache



memory management

memory en cache tonen

free informatie uit /proc/meminfo leesbaarder tonen

```
student@ubdesk:~$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	3,8G	1,2G	786M	14M	1,9G	2,3G
Swap:	2,0G	0B	2,0G			

-h human readable (megabytes, gigabytes)



memory management

memory en cache tonen

top toont ook informatie over memory

```
top - 15:52:06 up 54 min, 1 user, load average: 0,03, 0,04, 0,01
Tasks: 273 total, 1 running, 208 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3,4 us, 3,4 sy, 0,7 ni, 90,5 id, 1,8 wa, 0,0 hi, 0,1 si, 0,0 st
KiB Mem : 4015640 total, 804364 free, 1268832 used, 1942444 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 2444736 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
18650	student	20	0	51348	4240	3448	R	11,8	0,1	0:00.03	top
1	root	20	0	160256	9484	6620	S	0,0	0,2	0:04.55	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kthreadd



memory management

swap space

- als een OS meer geheugen nodig heeft dan aanwezig is via RAM
- is goedkoper, maar trager (want staat op harde schijf)
- kan een partitie zijn of een file of een combinatie

```
student@ubdesk:~$ free -h | grep -vi mem
```

	total	used	free	shared	buff/cache	available
Swap:	2,0G	0B	2,0G			

```
student@ubdesk:~$ cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/swapfile	file	2097148	0	-2

memory management

swap space

swap partitie aangemaakt

```
student@ubdesk:~$ sudo fdisk -l | grep /dev/sdb
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
/dev/sdb1          2048 4196351 4194304    2G 82 Linux swap / Solaris
```

mkswap

```
student@ubdesk:~$ sudo mkswap /dev/sdb1
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=41d2e15e-b15f-463e-85f1-e4137a5ead33
```

swapon en swapoff om swap te activeren of deactiveren

```
student@ubdesk:~$ sudo swapon /dev/sdb1
student@ubdesk:~$ swapon
```

NAME	TYPE	SIZE	USED	PRIO
/swapfile	file	2G	0B	-2
/dev/sdb1	partition	2G	0B	-3



memory management

swap space

```
student@ubdesk:~$ cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/swapfile	file	2097148	0	-2
/dev/sdb1	partition	2097148	0	-3

```
student@ubdesk:~$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	3,8G	1,1G	2,1G	7,7M	607M	2,5G
Swap:	4,0G	0B	4,0G			

```
student@ubdesk:~$
```



memory management

swapfile aanmaken

```
student@ubdesk:~$ sudo dd if=/dev/zero of=/swapfile2 bs=1024 count=2048000
2048000+0 records in
2048000+0 records out
2097152000 bytes (2,1 GB, 2,0 GiB) copied, 254,735 s, 8,2 MB/s
student@ubdesk:~$ sudo mkswap /swapfile2
mkswap: /swapfile2: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 2 GiB (2097147904 bytes)
no label, UUID=66bafcf4-4984-4b56-af93-aba2a128fe0d
student@ubdesk:~$ sudo chmod 0600 /swapfile2
student@ubdesk:~$ sudo swapon /swapfile2
student@ubdesk:~$ swapon
```

NAME	TYPE	SIZE	USED	PRIO
/swapfile	file	2G	0B	-2
/dev/sdb1	partition	2G	0B	-3
/swapfile2	file	2G	0B	-4

```
student@ubdesk:~$
```

memory management

swap space

in /etc/fstab (om te behouden na een reboot)

/dev/sdb1	none	swap	sw,pri=-1	0	0
/swapfile	none	swap	sw,pri=-2	0	0
/swapfile2	none	swap	sw,pri=-3	0	0

met swapon zie je ook de prioriteit van de swap spaces



<https://help.ubuntu.com/community/SwapFaq>

What is swap ?

Why do I need swap ?

How much swap do I need ?

How do I add more swap ?

...



System Management

Deel III : Logging

**DE HOGESCHOOL
MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Rsyslog-daemon

- rsyslogd
- Rocket-Fast **S**ystem for **L**og processing
- Versie kan je bekijken via
 - rsyslogd -v
 - Geeft 8.x.x

Rsyslog-daemon

- rsyslogd configuratie
 - /etc/rsyslog.conf
 - Modules, global directives, ...
 - `$IncludeConfig /etc/rsyslog.d/*.conf`
 - hierin staat dat default-logging-rules staan in `/etc/rsyslog.d/50-default.conf`
 - /etc/rsyslog.d/50-default.conf
 - hier zie je wat standaard gelogd wordt en naar welke log-file

Rsyslog-daemon

- Opmaak van een rsyslogd-configuratiebestand
 - facility.priority /path_to_logfile
 - facility
 - geeft het systeem aan dat het log-bericht heeft aangemaakt (=de origine)
 - bvb “kern” voor de messages afkomstig van de kernel
 - kan zijn: auth, authpriv, cron, daemon, kern, lpr, mail, news, syslog, user, uucp en local0-local7
 - alle logs van dit systeem met de aangegeven priority, en hogere, zullen gelogd worden naar de aangegeven file

Rsyslog-daemon

- facility.priority /path_to_logfile
- priority
 - geeft de ernst van het log-bericht aan
 - kan zijn: debug, info, notice, warning, err, crit, alert en emerg
 - vanaf de aangegeven priority en alle hogere zullen voor het aangegeven systeem(=facility) gelogd worden
- path_to_logfile
 - geeft aan naar welk bestand deze logs gestuurd worden

Rsyslog-daemon

- Voorbeelden voor rsyslogd-configuratiebestand
 - facility.priority /path_to_logfile
 - *.*;auth,authpriv.none -/var/log/syslog
 - alle log-berichten met eender welke prioriteit, behalve logberichten van auth en authpriv, worden gelogd naar /var/log/syslog
 - auth,authpriv.* /var/log/auth.log
 - alle logberichten van auth en authpriv worden gelogd naar /var/log/auth.log
 - mail.err /var/log/mail.err
 - logberichten van mail met een prioriteit hoger dan of gelijk aan err worden gelogd naar /var/log /mail.err

Rsyslog-daemon

- Voorbeelden voor rsyslogd-configuratiebestand
 - facility.priority /path_to_logfile
 - *.*;auth,authpriv.none -/var/log/syslog
 - wil zeggen alles behalve auth en authpriv loggen naar /var/log/syslog
 - “.” is scheiden van meerdere selectors
 - “,” is scheiden van meerdere facilities met dezelfde prioriteit
 - “-” voor het pad wil zeggen dat de kernel-buffer niet telkens moet geflushed worden als er een logbericht wordt weggeschreven. In de meeste distributies wordt sowieso niet geflushed en wordt het minteken dus ook niet in rekening gebracht.

Rsyslog-daemon

- Voorbeelden voor rsyslogd-configuratiebestand
 - facility.priority /path_to_logfile
 - kern.err /var/log/kern.log
 - alle log-berichten met als ernst meer of gelijk aan error
 - kern.!err /var/log/kern.log
 - alle log-berichten met als ernst minder dan “error”
 - kern.=info /var/log/kern.log
 - Alle log-berichten met als ernst het type “info”
 - kern.!=info /var/log/kern.log
 - alle log-berichten behalve die met als ernst het type “info”

Belangrijke logfiles

- `/var/log/syslog`
 - hiernaar wordt bijna alles gelogd door de syslog-daemon
 - het is de “General System Activity-log”
- `/var/log/auth.log`
 - bevat de user login en authorizations (bvb sudo-cmd's)

Belangrijke logfiles

- `/var/log/boot.log`
 - bevat de logs van de systeem-opstart-scripts van de vorige keren dat er geboot is
- `/var/log/kern.log`
 - bevat kernel-logs
- `/var/log/dmesg` (file bestaat blijkbaar niet meer, maar commando werkt nog)
 - bevat een dump van de kernel-message-buffer sinds boot
 - de kernel-logs van tijdens het booten kunnen ook bekeken worden met het commando “**dmesg**”

Belangrijke logfiles

- `/var/log/btmp`
 - bevat logs van mislukte login-pogingen
 - te bekijken via: `lastb` of `last -f /var/log/btmp | less`
- `/var/log/wtmp`
 - bevat records van logins en logouts, reboots en shutdowns
 - gebruik het commando “last” om deze file te tonen
 - om de laatste login-tijd te zien per gebruiker kan je ook het commando “lastlog” bekijken
 - “who” gebruikt deze file ook om te kijken wie ingelogd is

Belangrijke logfiles

- `/var/log/apport.log`
 - logfiles van crashes van applicaties
- `/var/log/dpkg.log`
 - bevat de logs van het Package-management-systeem
- `/var/log/<daemon>/....`
 - bepaalde daemons plaatsen hun logs in een subdir van `/var/log`

Logrotatie

- Sommige logfiles hebben een logrotatie
 - logrotate
 - zorgt voor automatische rotatie, verwijdering en compressie van logfiles
 - bvb: syslog , syslog.1 , syslog.2.gz , ... , syslog.7.gz
 - De algemene config-file is /etc/logrotate.conf
 - iedere daemon kan voor zijn log-rotatie een config-file aanmaken in /etc/logrotate.d/
 - daily, weekly of monthly of met een size van xxxMB
 - met een rotatie van n-files
 - met compressie, ...

Logfiles bekijken

- `cat /var/log/syslog`
- `more /var/log/syslog`
- `less /var/log/syslog`
- `grep -i dhclient /var/log/syslog`
- `tail -f /var/log/syslog`
 - -f van follow
 - toont onmiddellijk de laatste 10 regels van de logfile
 - houdt de file open en toont ook de nieuwe log-berichten als de logfile groeit

Zelf iets loggen

- Zelf iets naar syslog sturen kan met
 - het commando “logger”
 - logger “Gebruiker \$USER ingelogd”
 - logger met de optie -t
 - hiermee kan je aangeven van welk script of daemon het logbericht komt
 - logger -t kernel “Couldn’t find apache-module”