

Inhoud

Hoofdstuk 6 Onderhoud van systeemopstart en services in Linux (NL)	2
Het init-systeem en systemd	2
Werken met systemd	3
Unit-bestanden en servicebeheer	3
Servicebeheer met systemctl	3
Beheer van opstartdoelen (targets)	4
Systeemanalyse met systemd-analyze	4
Werken met SysV init	5
Uitgebreide samenvatting van de examenonderdelen	5
Beschrijf het init-programma	6
Samenvatting van systemd-units	6
Uitleg over systemd-service units	6
Hoe beheert men SysV init met commando's?	6
Beschrijf systemd mount- en automount-unitbestanden	6
Chapter 6 Maintaining System Startup and Services in Linux (EN)	7
The Init System and systemd	7
Working with systemd	8
Unit Files and Service Management	8
Service Management with systemctl	8
Managing Startup Targets	9
System Analysis with systemd-analyze	9
Working with SysV init	10
Comprehensive Exam Summary	10
Describe the init program	10
Overview of systemd Units	11
Understanding systemd Service Units	11
How to Manage SysV init with Commands	11
Describe systemd Mount and Automount Unit Files	11

Hoofdstuk 6 Onderhoud van systeemopstart en services in Linux (NL)

Na het opstartproces komt je Linux-systeem in de fase van de definitieve systeeminitialisatie, waarin verschillende services (ook wel daemons genoemd) worden gestart. Dit zijn programma's die specifieke taken uitvoeren, zoals netwerkbeheer, tijdsynchronisatie en beveiliging. Dit hoofdstuk behandelt hoe services worden gestart, beheerd en geoptimaliseerd met behulp van systemd, het moderne init-systeem dat de oudere SysV init heeft vervangen.

Het init-systeem en systemd

Het init-programma (PID 1) is het ouderproces van alle andere services op een Linux-systeem. Afhankelijk van de distributie wordt een van de volgende methoden gebruikt voor systeeminitialisatie:

- **SysV init:** Het klassieke init-systeem, dat gebaseerd is op Unix System V en werkt met runlevels om services te starten.
- **systemd:** Een modern init-systeem dat parallelle service-opstart mogelijk maakt en flexibele configuraties biedt.

Verschil tussen SysV init en systemd:

- SysV init gebruikt runlevels (0-6) om te bepalen welke services worden gestart.
- systemd gebruikt **targets**, wat groeperingen zijn van services, zoals **multi-user.target** en **graphical.target**.
- SysV init werkt met shellscripts in **/etc/init.d/** om services te starten en te stoppen.
- systemd gebruikt **unit-bestanden** om services te definiëren, die efficiënter en sneller zijn.

Een systeem controleren op het gebruikte init-systeem:

`which init`

Controleer of systemd actief is:

`readlink -f /sbin/init`

Als de uitvoer verwijst naar `/usr/lib/systemd/systemd`, dan wordt systemd gebruikt.

Controleer de PID 1-procesnaam:

`ps -p 1`

Als de uitvoer `systemd` toont, gebruikt het systeem systemd.

Werken met systemd

Unit-bestanden en servicebeheer

systemd beheert services via unit-bestanden die zich bevinden in `/etc/systemd/system/` of `/usr/lib/systemd/system/`. De belangrijkste unit-types zijn:

- **Service units** (`.service`): Definiëren individuele services.
- **Target units** (`.target`): Groeperingen van services.
- **Mount units** (`.mount`): Automatische koppeling van bestanden.
- **Timer units** (`.timer`): Geplande taken, vergelijkbaar met `cron`.

Een lijst van actieve units bekijken:

```
systemctl list-units
```

Een lijst van alle unit-bestanden bekijken:

```
systemctl list-unit-files
```

Servicebeheer met systemctl

Een service starten:

```
systemctl start servicenaam
```

Een service stoppen:

```
systemctl stop servicenaam
```

Een service herstarten:

```
systemctl restart servicenaam
```

Zorg ervoor dat een service bij het opstarten wordt gestart:

```
systemctl enable servicenaam
```

Voorkom dat een service bij het opstarten wordt gestart:

```
systemctl disable servicenaam
```

De status van een service controleren:

```
systemctl status servicenaam
```

Controleren of een service actief is:

`systemctl is-active servicenaam`

Mislukte services bekijken:

`systemctl --failed`

Beheer van opstartdoelen (targets)

`systemd` gebruikt targets in plaats van runlevels.

Het huidige target bekijken:

`systemctl get-default`

Het standaard target wijzigen:

`systemctl set-default multi-user.target`

Direct overschakelen naar een nieuw target:

`systemctl isolate multi-user.target`

Speciale targets:

- `rescue.target`: Beperkte modus voor probleemoplossing.
- `emergency.target`: Alleen root-toegang, minimale services actief.
- `poweroff.target`: Schakelt het systeem uit.
- `reboot.target`: Herstart het systeem.

Direct naar rescue-modus schakelen:

`systemctl isolate rescue.target`

Systeemanalyse met systemd-analyze

De tool `systemd-analyze` helpt bij het diagnosticeren van opstartproblemen.

De opstarttijd bekijken:

`systemd-analyze time`

Bekijken welke services de meeste vertraging veroorzaken:

`systemd-analyze blame`

Unit-bestanden controleren op fouten:

systemd-analyze verify

De kritieke paden in het opstartproces tonen:

systemd-analyze critical-chain

Werken met SysV init

Op oudere systemen die SysV init gebruiken, wordt de standaard runlevel ingesteld in `/etc/inittab`.

De huidige en vorige runlevel bekijken:

runlevel

De status van alle services bekijken:

service --status-all

Een service starten:

service servicenaam start

Een service stoppen:

service servicenaam stop

Zorg ervoor dat een service bij het opstarten wordt gestart:

chkconfig servicenaam on

Direct overschakelen naar een ander runlevel:

init 3 # Multi-user modus zonder GUI

init 5 # Multi-user modus met GUI

Hoewel SysV init nog steeds in sommige systemen voorkomt, wordt het steeds vaker vervangen door systemd.

Uitgebreide samenvatting van de examenonderdelen

Beschrijf het init-programma

Het init-programma is het eerste proces op een Linux-systeem (PID 1).

Op moderne systemen is systemd de standaard en init is vaak een symbolische link naar `/usr/lib/systemd/systemd`.

Samenvatting van systemd-units

systemd gebruikt unit-bestanden om services en processen te beheren.

Belangrijke unit-types: `.service`, `.target`, `.mount`, `.automount`.

Een lijst van actieve units bekijken:

```
systemctl list-units
```

Uitleg over systemd-service units

Services worden beheerd via `.service-unit`-bestanden.

Een service-unit bevat drie secties: `[Unit]`, `[Service]`, `[Install]`.

Een service starten:

```
systemctl start servicenaam
```

Hoe beheert men SysV init met commando's?

Overstappen naar runlevel 3:

```
init 3
```

Een service starten:

```
service servicenaam start
```

De status van alle services tonen:

```
service --status-all
```

Beschrijf systemd mount- en automount-unitbestanden

Mount units (`.mount`) beheren bestandssysteemkoppelingen.

Automount units (`.automount`) zorgen voor on-demand koppeling.

Timer units (`.timer`) vervangen cron-taken.

Chapter 6 Maintaining System Startup and Services in Linux (EN)

After the boot process, your Linux system enters the final system initialization phase, where various services (also called daemons) are started. These are programs that perform specific tasks, such as network management, time synchronization, and security. This chapter covers how services are started, managed, and optimized using `systemd`, the modern init system that has replaced the older SysV init.

The Init System and `systemd`

The `init` program (PID 1) is the parent process of all other services on a Linux system. Depending on the distribution, one of the following methods is used for system initialization:

- **SysV init:** The classic init system, based on Unix System V, which uses runlevels to determine which services are started.
- **`systemd`:** A modern init system that enables parallel service startup and offers flexible configurations.

Difference between SysV init and `systemd`:

- SysV init uses **runlevels** (0-6) to determine which services are started.
- `systemd` uses **targets**, which are groups of services, such as **`multi-user.target`** and **`graphical.target`**.
- SysV init works with shell scripts in `/etc/init.d/` to start and stop services.
- `systemd` uses **unit files**, which define services more efficiently and start them faster.

Check which init system your Linux distribution is using:

`which init`

Verify if `systemd` is active:

`readlink -f /sbin/init`

If the output refers to `/usr/lib/systemd/systemd`, then the system is using `systemd`.

Check the process name for PID 1:

`ps -p 1`

If the output shows `systemd`, the system is running `systemd`.

Working with systemd

Unit Files and Service Management

systemd manages services through unit files located in `/etc/systemd/system/` or `/usr/lib/systemd/system/`. The main unit types are:

- **Service units** (`.service`): Define individual services.
- **Target units** (`.target`): Groups of services.
- **Mount units** (`.mount`): Automatically mounting filesystems.
- **Timer units** (`.timer`): Scheduled tasks, similar to `cron`.

List active units:

```
systemctl list-units
```

List all unit files:

```
systemctl list-unit-files
```

Service Management with systemctl

Start a service:

```
systemctl start service-name
```

Stop a service:

```
systemctl stop service-name
```

Restart a service:

```
systemctl restart service-name
```

Enable a service at boot:

```
systemctl enable service-name
```

Disable a service at boot:

```
systemctl disable service-name
```

Check the status of a service:

```
systemctl status service-name
```

Check if a service is running:

```
systemctl is-active service-name
```


View failed services:

```
systemctl --failed
```

Managing Startup Targets

systemd uses targets instead of runlevels.

View the current target:

```
systemctl get-default
```

Change the default target:

```
systemctl set-default multi-user.target
```

Switch immediately to a new target:

```
systemctl isolate multi-user.target
```

Special targets:

- `rescue.target`: Limited mode for troubleshooting.
- `emergency.target`: Only root access, minimal services active.
- `poweroff.target`: Shuts down the system.
- `reboot.target`: Reboots the system.

Switch to rescue mode:

```
systemctl isolate rescue.target
```

System Analysis with systemd-analyze

The `systemd-analyze` tool helps diagnose startup issues.

View boot time:

```
systemd-analyze time
```

See which services cause the most delay:

```
systemd-analyze blame
```

Check unit files for errors:

systemd-analyze verify

Show the critical paths in the boot process:

systemd-analyze critical-chain

Working with SysV init

On older systems using SysV init, the default runlevel is set in `/etc/inittab`.

View the current and previous runlevel:

runlevel

Check the status of all services:

service --status-all

Start a service:

service service-name start

Stop a service:

service service-name stop

Ensure a service starts at boot:

chkconfig service-name on

Switch immediately to a different runlevel:

init 3 # Multi-user mode without GUI

init 5 # Multi-user mode with GUI

Although SysV init is still present in some systems, it is increasingly being replaced by systemd.

Comprehensive Exam Summary

Describe the init program

The init program is the first process on a Linux system (PID 1).
On modern systems, systemd is the default, and init is often a symbolic link to `/usr/lib/systemd/systemd`.

Overview of systemd Units

systemd uses unit files to manage services and processes.
Key unit types: `.service`, `.target`, `.mount`, `.automount`.

List active units:

```
systemctl list-units
```

Understanding systemd Service Units

Services are managed through `.service` unit files.
A service unit contains three sections: `[Unit]`, `[Service]`, `[Install]`.

Start a service:

```
systemctl start service-name
```

How to Manage SysV init with Commands

Switch to runlevel 3:

```
init 3
```

Start a service:

```
service service-name start
```

Show the status of all services:

```
service --status-all
```

Describe systemd Mount and Automount Unit Files

Mount units (`.mount`) manage filesystem mounting.
Automount units (`.automount`) enable on-demand mounting.
Timer units (`.timer`) replace cron jobs.