

Lektionstillfälle 13

Processer och demoner
med Mikael Larsson

Återblick

Tidigare i veckan ägnade vi oss åt att skripta och manipulera textfiler. Vi sökte i loggfiler, skapade rapporter och gjorde enklare beräkningar.

Hur gick det?

Dagens lektion

Mål: Att kunna starta och stoppa tjänster samt ändra schemaläggning och förstå uppstart

- Bootstrapping
- Systemtjänster
 - systemctl, service
- Schemalagda aktiviteter
 - crontab, systemd
- Runlevels

Tid för frågor och upphämtning av tidigare labbar.

Termer och begrepp

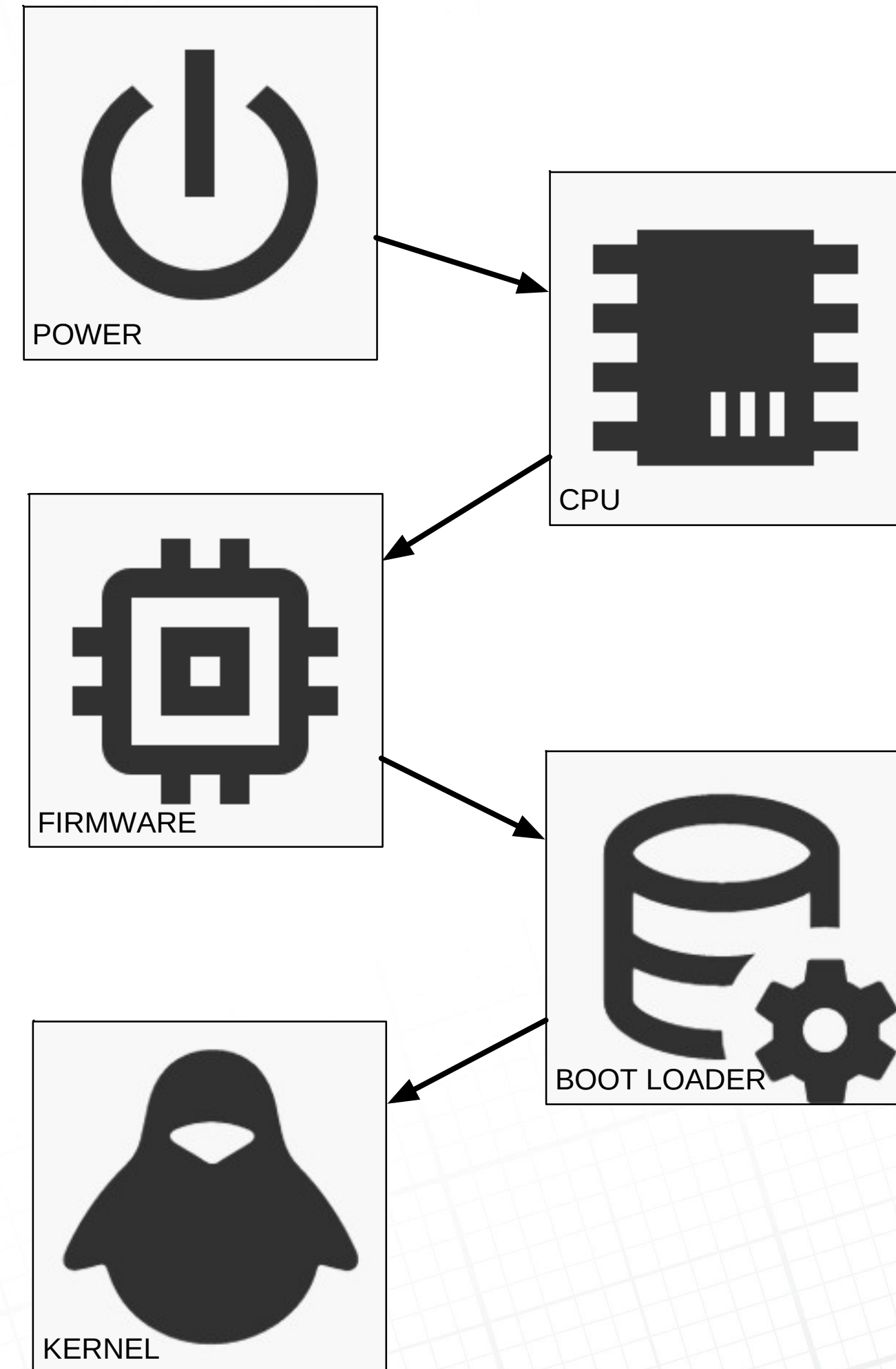
- tjänst / service
- demon / daemon
- cron, crontab
- systemd
- unit, target, runlevel

Bootstrapping, att boota

? Varför heter det "boota"?

Vi pratade om uppstart när vi installerade Linux
lektion 2.

Det vi kommer titta närmre på idag är stegen efter
"kernel" i bilden.



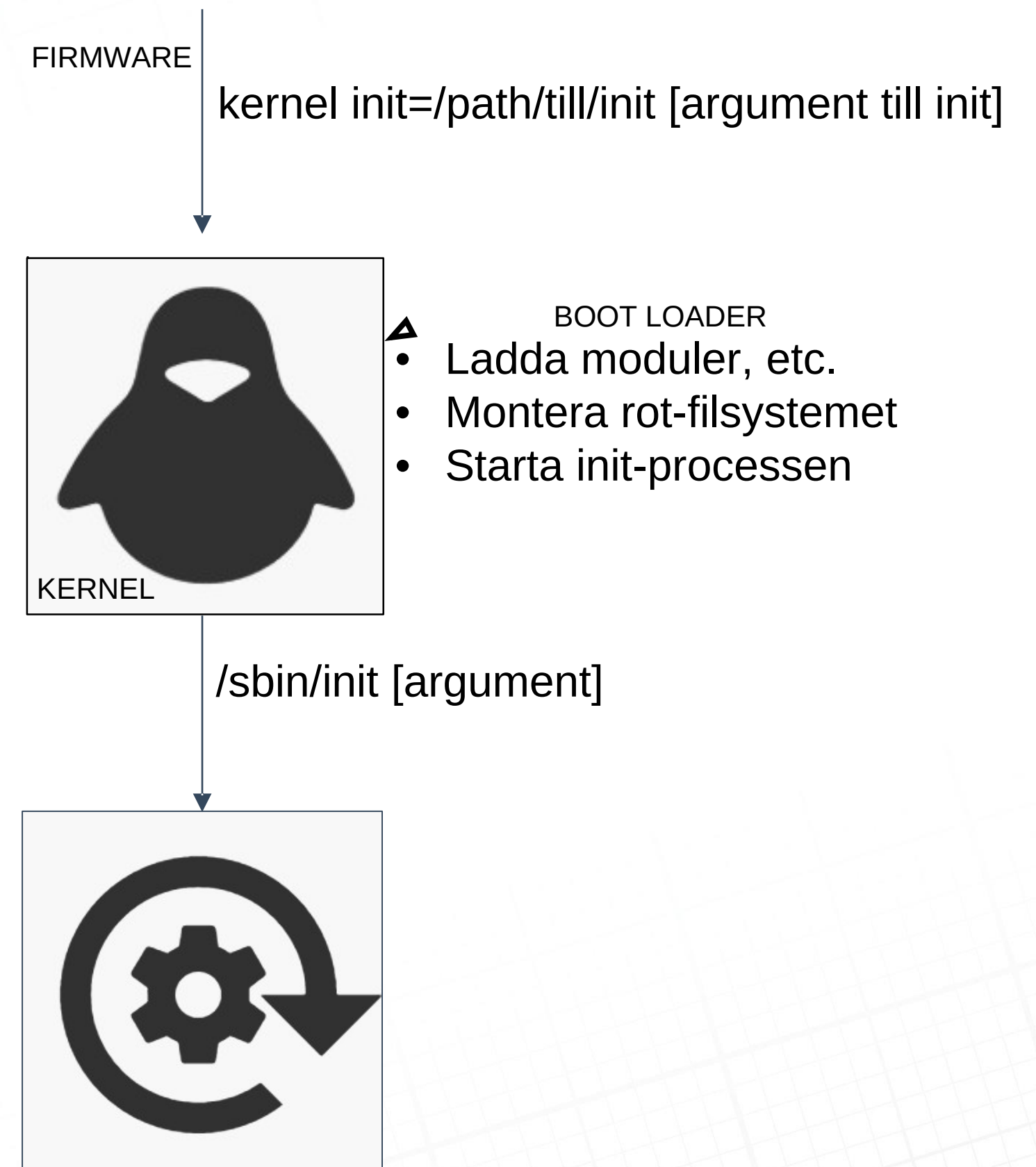
Den första processen

Uppstart, steg för steg:

- Kärnan laddas
- De moduler som krävs för vidare uppstart laddas
- Rotfilsystemet monteras
- Den första processen "init" startas och får PID 1

Vilket program som ska agera "init" kan anges som argument till kärnan.

Man kan alltså göra en speciell installation som använder en helt egen uppstartsprocess om man vill.

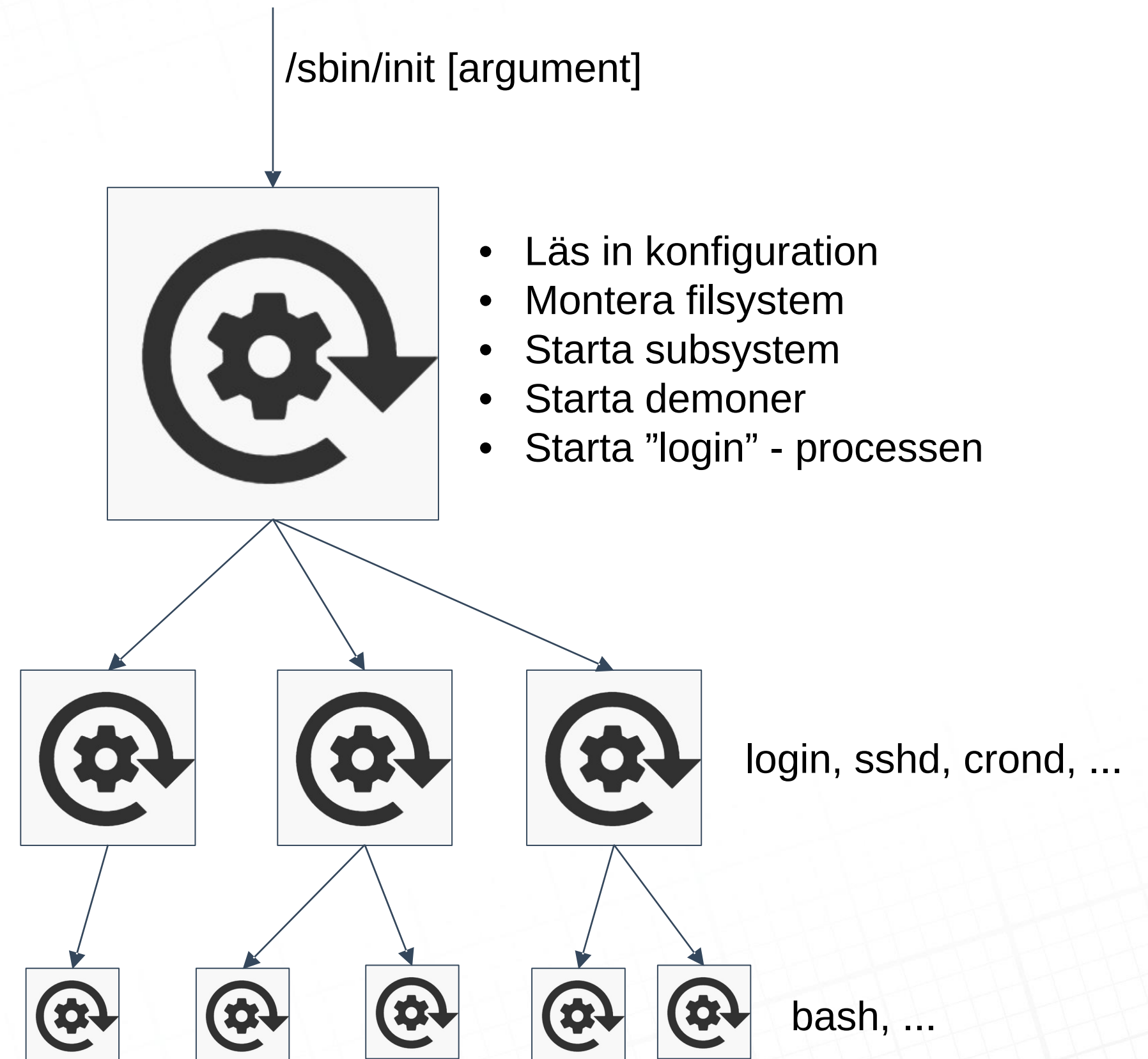


Vad gör "init"?

När init startar går den igenom följande steg:

- Läser konfiguration från filer i /etc
- Monterar filsystem
- Startar subsystem för nätverk, meddelandehantering, et.c.
- Startar demoner
- Startar "login" – processen för interaktiv inloggning

Observera att init äger hela processträdet och körs så länge systemet kör.



Men vad är "systemd"?

System-demonen, "**systemd**", är den version av "init" som används av Ubuntu och andra distros.

Den ersätter "UNIX System V init" som varit standard under lång tid.

"**systemd**" har också funktionalitet inbyggd som tidigare var utspridd på flera olika demoner och koncept:

- Schemaläggning (tidigare crond)
- Loggning (tidigare syslogd)
- Inloggning (tidigare logind)

Men – övergången till de här inbyggda funktionerna pågår fortfarande. Det är många existerande program som behöver anpassas och det tar tid.

Runlevels och targets

”runlevel” är ett koncept från ”System V init” som anger vilken servicenivå ett system körs på.

I ”systemd” heter motsvarande koncept ”target”.

En stor skillnad mellan **runlevel** och **target** är att endast **en runlevel** kan vara aktiv, medan **flera targets** kan vara aktiva samtidigt.

Från ”runlevel” – manbladet:

Runlevel	Target
0	poweroff.target
1	rescue.target
2, 3, 4	multi-user.target
5	graphical.target
6	reboot.target

Default target och konfiguration

"systemd" bor i "/lib/systemd", se hur "init" pekar dit:

```
frasse@frasse:/lib/systemd$ ls -l /sbin/init  
lrwxrwxrwx 1 root root 20 Jan  6 21:04 /sbin/init -> /lib/systemd/systemd
```

Konfigurationen ligger i "/etc/systemd", som mest refererar till komponenter, "units", som ligger i "/lib/systemd/system".

Varje "target" är en sådan unit och motsvaras av en fil med ändelsen ".target".

Systemtjänster och demonprocesser

Varje systemtjänst är en "unit" i **systemd**-världen och definieras av en fil med ändelsen ".service".

Exempelvis definieras tjänsten "ssh" av filen "ssh.service". Några rader i filen bestämmer hur tjänsten startas:

```
[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
```

Vi kan se att SSH-demonen "/usr/bin/sshd" anropas för att utföra olika steg.

En "demonprocess" – daemon, är en process som körs fristående från en inloggad användare.

Kommandon

Target-kommandon:

- `systemctl list-units --type target`
- `systemctl get-default`
- `systemctl set-default`

Service-kommandon:

- `systemctl [start | status | reload | stop]`

Laboration

Utforska "systemd" – systemet!
Se instruktion i portalen.



Schemaläggning med cron

”cron” är den demon som hanterar schemalagd exekvering av kommandon.

Grundkonfigurationen för vilka jobb som ska köras utgörs av en ”crontab” – en fil på ett speciellt format. Varje användare har en egen ”crontab”.

Systemets ”crontab” ägs av root.

crontab är även kommandot för att lista och redigera ”crontab”.

”crontab -l” visar tabellen

”crontab -e” redigerar tabellen

crontab

Använd "crontab -e" för att skapa eller redigera din crontab.

Varje rad har följande fält:

min hour dom mon dow command

Fälten före "command" kan vara ett tal, en lista med tal, ett intervall eller en asterisk.

Exempel:

5 21-23 * * 0,6 /usr/local/bin/my-report

Raden kör "my-report" fem minuter över varje hel timme mellan 21 och 23 på lördagar och söndagar.

Man kan också ange var n:te genom att lägga till /n efter ett tal. Exempelvis */5 i minut-fältet anger var femte minut.

<https://crontab.guru>

Enklare crontab

Istället för den krångliga syntaxen kan man i crontab ange ett symboliskt värde, exempelvis "@daily" eller "@weekly":

```
@daily /usr/local/bin/my-report
```

Raden kör "my-report" en gång om dagen.

Man kan också lägga in cron-jobb för root genom att lägga skript i "/etc/cron.daily" eller syskonkatalogerna.

OBS – resultatet av ett cron-jobb (utskrifter) mailas till ägaren. Det är opraktiskt, så man brukar se till att cron-jobb istället loggar till fil och aldrig producerar någon utskrift.

crontab - loggning

Notera att det är skillnad på vad själva cron-jobbet skriver ut och vad cron-demonen loggar!

```
When executing commands, any output is mailed to the owner of the crontab (or to the user named in the MAILTO environment variable in the crontab, if such exists). The children copies of cron running these processes have their name coerced to uppercase, as will be seen in the syslog and ps output.
```

`-L loglevel`

```
Tell cron what to log about jobs (errors are logged regardless of this value) as the sum of the following values:
```

- 1 will log the start of all cron jobs
- 2 will log the end of all cron jobs
- 4 will log all failed jobs (exit status != 0)
- 8 will log the process number of all cron jobs

```
The default is to log the start of all jobs (1). Logging will be disabled if levels is set to zero (0). A value of fifteen (15) will select all options.
```


Schemaläggning med systemd

"systemd" har en unit-typ, "timer" som kan användas istället för "cron".

En "timer" – unit triggar körning av en "service" – unit. De har en mycket rikare syntax än "cron" och kan till exempel justera start-tiden slumpmässigt för att sprida last.

Se exempelvis motd-news.timer:

```
[Unit]
Description=Message of the Day

[Timer]
OnCalendar=00,12:00:00
RandomizedDelaySec=12h
Persistent=true
OnStartupSec=1min

[Install]
WantedBy=timers.target
```

Laboration

Experiment med crontab.
Se instruktion i portalen!

Summering

Idag har vi tittat på systemstart och vad som händer från det att kärnan laddas till att man kan logga in.

Vi har tittat på vilken roll "systemd" har, och hur man kan använda "cron" för att schemalägga aktiviteter.

Nästa gång

Mål: Att arbeta med filarkiv, göra backup lokalt och till molntjänst

- tar, gzip, zip
- rsync
- rclone

Stort tack!