

Het boot proces



In dit hoofdstuk maken we nader kennis met het Linux boot proces: van poweron naar de login prompt

Doelstelling

Aan het eind van dit hoofdstuk is de cursist bekend met:

- de rol van firmware -> boot loader -> kernel
- boot loaders en hoe er mee om te gaan
- het gehele boot proces
- runlevels (en hoe een server netjes uit te zetten)
- nieuwere technieken (niet in detail): upstart en systemd
- de bij bovenstaande acties behorende commando's

boot loaders (1/2)

De bootloader is de schakel tussen het firmware en de kernel

- Het firmware (BIOS/EFI) leest en execute de boot loader
- De boot loader laad de kernel in memory en start deze
- Met het BIOS is de boot-loader te vinden in het Master Boot Record (MBR, de eerste 512 bytes van de hard disk).
 - In het geval van GUID Partitioning (GPT) met Logical BLock Addressing (LBA) is er een legacy MBR in de eerste LBA (LBA-0)
- Met EFI wordt voor de boot-loader verwezen naar een file op een ESP partitie
- De boot loader is te configureren. Dit kunt u bv. doen om:
 - met speciale opties te booten
 - een ander OS te booten

boot loaders (2/2)

We onderschieden drie bootloaders:

- LILO de Linux loader. Verouderd en vervangen door GRUB of GRUB2
- GRUB de Grand Unified Boot Loader (tegenwoordig "legacy GRUB")
 - Deze kan niet altijd overweg met het "legacy MBR" bij GPT partitonering of met een EFI bootloader-file.
- GRUB2 de opvolger van GRUB

MBR

Het MBR bevat de (primary) boot loader code en de partitie tabel

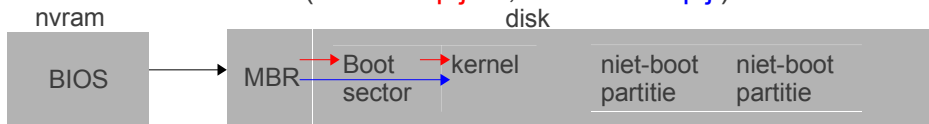
De **primary** bootloader doet of:

- 1) kijkt in de partitietabel naar een bootable partitie
laad van die partitie de bootsector en runt deze
deze bootsector bevat een tweede (**secondary**) bootloader welke de kernel weet te laden en te starten

Of de primary boot loader doet:

- 2) laad de kernel direct, zonder tweede bootloader, en start deze

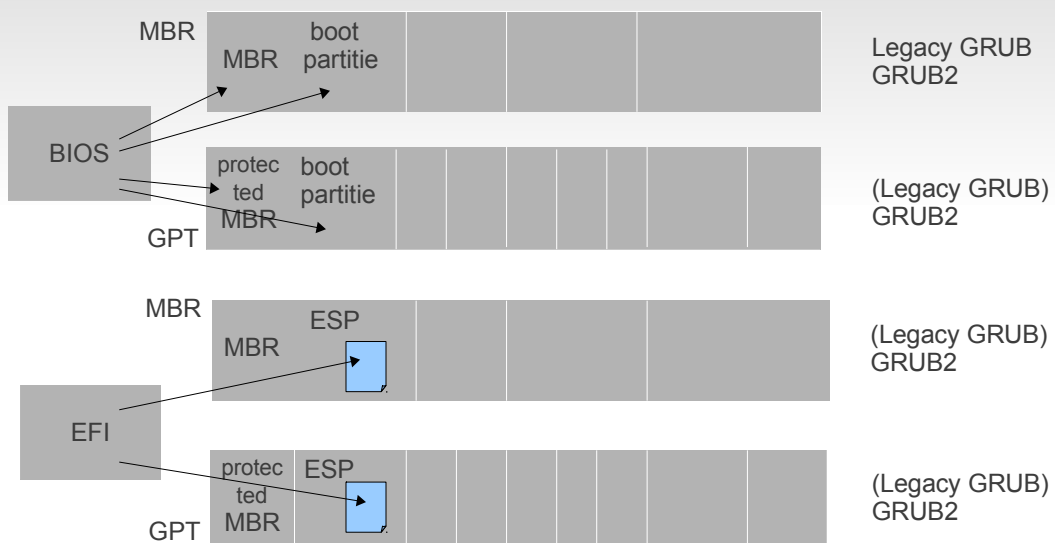
Zie onderstaand schema (1 = rode pijlen, 2 = blauwe pijl):



Windows boot komt redelijk overeen met optie 1. Linux is veel flexibeler: de boot loader hoeft ook niet in het MBR, kan ook alleen in de bootsector van de bootpartitie.

Firmware, partitionering en boot loaders

Samengevat in een plaatje:



MBR of partitie boot sector?

De linux boot loader kun je installeren in het MBR of in de bootsector van de boot partitie.

De voor- en nadelen op een rijtje:

- Bootloader in MBR: kan overschreven worden door een windows install (bij bv. een dual boot)
- In boot partitie: de windows boot loader in het MBR kan deze negeren (dan is het nodig om je Linux partitie als *active* te markeren met fdisk)

Voorkeur: maakt weinig uit maar pas op voor overschrijven!

Linux only installaties: MBR

Voor Linux hoeft de boot-partitie niet een primary partitie te zijn (maar is wel gebruikelijk).

Tot nu toe: een wat simpele voorstelling van zaken. Boot loaders zijn best complex:

- mogelijkheid tot het tonen van een menu om te kiezen uit verschillende OS-en en/of kernel versies
- koppelen van boot loaders (chaining) zodat er meerdere bootloaders op 1 partitie gebruikt kunnen worden

de GRUB boot loader

Meer features dan LILO:

- geen re-install in boot sector nodig na configuratie wijziging (GRUB leest configuratie tijdens het booten)
- meer interactieve boot opties

Config in:

`/boot/grub/menu.lst` of `/boot/grub/grub.conf` (Fedora, RedHat)

Soms symlink `/etc/grub.conf` naar bovenstaande files.

overzicht grub config

```
default 1
timeout 10

title Windows XP
    rootnoverify (hd0,1)
    chainloader +1 (laad andere (2nd) boot loader (sector 1 van root))

title Ubuntu, kernel 2.6.14-10-generic
    root (hd0,4)
    kernel /vmlinuz-2.6.14-10-generic root=/dev/sda1 ro quiet splash
    initrd /initrd.img-2.6.14-10-generic
    quiet

title Ubuntu, kernel 2.6.14-10-generic (recovery mode)
    root (hd0,4)
    kernel /vmlinuz-2.6.14-10-generic root=/dev/sda1 ro single
    initrd /initrd.img-2.6.14-10-generic

title Ubuntu, memtest86+
    root (hd0,4)
    kernel /memtest86+.bin
    quiet
```

GRUB install

Zoals gezegd, in tegenstelling tot LILO hoeft je GRUB na configuratie wijzigingen niet opnieuw te installeren.

Er zijn wel situaties te bedenken wanneer een GRUB install wel nodig is, bv.:

- bij verplaatsen van je boot partitie
- je installatie overzetten op een andere disk
- GRUB verplaatsen van het MBR naar een boot partitie (of andersom)

Commando's:

```
grub-install /dev/sda of grub-install '(hd0) '
grub-install /dev/sda1 of grub-install '(hd0,0) '
```

GRUB interactie

Tijdens het booten verschijnt een menu

Om boot opties te wijzigen:

- Ga naar de juiste kernel (met de pijltjes toetsen)
- Om boot parameters te wijzigen: druk op `e`
- Ga naar de regel met de kernel opties
- Druk weer op `e`
- Verander bv.:
 - voeg `1` toe om te booten in single user mode
 - voeg `init=/bin/bash` toe om direct een root-shell te krijgen (handig bij root password recovery!)
- Wanneer je klaar bent druk op `<Enter>`
- Druk op `b` voor booten

de GRUB2 boot loader

GRUB was niet meer onderhoudbaar, GRUB2 is geheel opnieuw gebouwd

GRUB2 wordt tegenwoordig ook gewoon GRUB genoemd (met versie nummer)

Initiële install ook met `grub-install`. Config in `/boot/grub/grub.cfg`

De config aanpassen met:

- direct editen van `/boot/grub/grub.cfg`
- met files in `/etc/grub.d` (met name `/etc/grub.d/40_custom`)
- met `/boot/grub/custom.cfg`

Run `grub-mkconfig` (of `update-grub`) na wijzigingen in `/boot/grub/custom.cfg` en/of files onder `/etc/grub.d` om een nieuwe `/boot/grub/grub.cfg` te genereren. **Dit overschrijft de directe edits!**

File `/etc/default/grub` bepaald "gedrag" van `grub-mkconfig`

Partitie nummering begint bij 1 (i.p.v. 0)

Gebruik van dynamisch te laden modules (`/boot/grub/*.mod`)

Ook op andere, niet PC BIOS, platformen beschikbaar (SPARC, PowerPC, ...)

deel grub2 config

```
menuentry 'Ubuntu, with Linux 3.2.0-41-generic-pae' --class ubuntu
--class gnu-linux --class gnu --class os {
    recordfail
    gfxmode $linux_gfx_mode
    insmod part_msdos
    insmod ext2
    set root='(hd0,msdos1)'
    linux    /vmlinuz-3.2.0-41-generic-pae root=UUID=6c59aed0-16fa-
4e5f-9d8a-a290ca76a9df ro
    initrd   /initrd.img-3.2.0-41-generic-pae
}
menuentry 'Ubuntu, with Linux 3.2.0-41-generic-pae (recovery mode)'
--class ubuntu --class gnu-linux --class gnu --class os {
    recordfail
    insmod part_msdos
    insmod ext2
    set root='(hd0,msdos1)'
    echo     'Loading Linux 3.2.0-41-generic-pae ...'
    linux    /vmlinuz-3.2.0-41-generic-pae root=UUID=6c59aed0-16fa-
4e5f-9d8a-a290ca76a9df ro recovery nomodeset
    echo     'Loading initial ramdisk ...'
    initrd   /initrd.img-3.2.0-41-generic-pae
}
```

het boot proces

Het boot proces in stappen:

- power
- CPU runt code in het BIOS
- BIOS checkt (POST) / initialiseert hardware, laad de boot loader
- de boot loader laad de kernel in geheugen en start deze
- de kernel:
 - initialiseert randapparatuur (devices)
 - mount de root partitie
 - runt /sbin/init
- /sbin/init brengt het systeem in het juiste runlevel (bv. single user mode, multi-user mode, multi-user grafische mode)
- login prompt wordt getoond (door /bin/login)

init en runlevels

De kernel runt /sbin/init. /sbin/init kijkt in /etc/inittab. Een voorbeeld:

```
id:2:initdefault:
si::sysinit:/etc/init.d/rcS
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6
z6:6:respawn:/sbin/sulogin
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
1:2345:respawn:/sbin/getty 38400 tty1
```

elke regel bevat 4 velden gescheiden door een dubbele punt:

1e veld: label (/etc/inittab wordt niet sequentieel doorlopen)

2e veld: runlevel

3e veld: actie (sysinit, initdefault, wait, respawn)

4e veld: te runnen commando

runlevels (1/2)

Een Linux systeem boot in een bepaald runlevel.

Elk runlevel heeft zijn specifieke eigenschappen:

runlevel	doel	functionaliteit
0	shutdown (poweroff)	geen
1, s of S	single user runlevel Voor maintainanc e	geen mounts, geen netwerk, geen user logins
2	multi user	Veelal niet gebruikt. Op debian/ubuntu: multi user, tekst of grafisch
3	multi user	Veel distro's: multi user, text-login
4	-	beschikbaar voor speciale aanpassingen
5	multi user	Veel distro's: multi user, grafisch
6	reboot	reboot

runlevels (2/2)

Het runlevel kun je zien met het command `runlevel`.

- dit commando geeft het vorige en huidige runlevel
- indien er geen vorig runlevel is wordt een N getoond

```
$ runlevel
```

```
N 2
```

Met `who -r` kan ook het huidige runlevel verkregen worden

De rc scripts

rc-scripts voor runlevel 2 in `/etc/rc2.d`

rc-scripts voor runlevel 3 in `/etc/rc3.d`

...

Per directory `/etc/rcX.d` wordt gebruik gemaakt van "Kill" en "Start" files

- "Kill" files: beginnen met een K en killen alles wat niet thuis hoort in dat runlevel
- "Start" files beginnen met een S en starten alles wat thuis hoort in dat runlevel

Een voorbeeld van de inhoud van `/etc/rc1.d` op een Debian 6 server:

```
$ ls /etc/rc1.d/
```

```
K01munin-node K02bacula-fd K02postfix K05rsyslog S01bootlogs  
S02single K02apache2 K02nagios-nrpe-server K03mysql  
S01killprocs
```

Files in `/etc/rcX.d` zijn symbolic links naar de echte files onder `/etc/init.d`

Upstart (1/2)

Het init-model met `/etc/inittab` en rc-scripts kom je steeds minder tegen ten faveure van Upstart (en systemd).

De eigenschappen van Upstart:

- event driven
- geen `/etc/inittab` meer
- processen kunnen simultaan starten: kortere boot tijd
- oorspronkelijke init-scripts runnen alleen bij verandering van runlevel, upstart is veel flexibeler (bv. in omgaan met "removable hardware" als bv. usb devices, mounten van netwerk devices): **meer "hotplug" bestendig**
- backwards compatible met init voor die services die nog niet met upstart geïmplementeerd zijn.
- scripts (jobs) in `/etc/init` (niet `/etc/init.d`, deze is alleen voor backwards compatibility)
- terminologie: jobs en events

Upstart (2/2)

Voorbeeld van een upstart job:

```
root@cws001:/etc/init# cat udev.conf
# udev - device node and kernel event manager
#
# The udev daemon receives events from the kernel about changes in the
# /sys filesystem and manages the /dev filesystem.

description    "device node and kernel event manager"

start on virtual-filesystems
stop on runlevel [06]

expect fork
respawn

exec /sbin/udev --daemon
root@cws001:/etc/init#
```

Commando's

`initctl stop udev; start udev` (of `reload udev`)

systemd

Snel opkomend boot systeem.

- net als Upstart niet alleen voor booten maar event-driven en goed met hotplug devices.
- configuratie veelal onder `/etc/systemd`
- ook backwards compatible met init voor die services die nog niet met systemd geïmplementeerd zijn.
- services stoppen/starten met de `systemctl` utility
- terminologie: units (services)

Enkele voorbeelden met `systemctl`:

- `systemctl list-units`
- `systemctl {stop|start|reload|restart|status|} unit`
- `systemctl {enable|disable} unit`

shutdown

Tijd om onze systemen down te brengen.

```
shutdown -h now
```

Met netjes wordt bedoeld dat alle init scripts doorlopen worden om alle processen te beindigen.

alternatieven:

- `poweroff`
- `sync, sync, sync, halt`

Geen goed idee om gewoon de powerknop in te drukken. Overigens zal de acpi daemon (`acpid`), mits actief natuurlijk, er dan alsnog voor zorgen dat er een nette shutdown gebeurt.

Referenties

http://en.wikipedia.org/wiki/GUID_Partition_Table

<http://www.gnu.org/software/grub/manual/grub.html>

<http://upstart.ubuntu.com/cookbook>

<http://www.freedesktop.org/wiki/Software/systemd>

Oefeningen

Tijd voor oefening!
