

Inhoud

Het Opstartproces van een Linux-systeem	2
Stappen in het Opstartproces	2
Bewaken en Probleemoplossing van het Opstartproces	4
Bootloader Configuratie en Herstel.....	4
Systeemherstel Technieken	5
Examen Essentials	5
Linux Bootproces:	5
Linux GRUB Legacy en GRUB2 Bootloaders:	6
Alternatieve Linux Bootloaders:	7
Herstellen van een Kernel Panic:	8
Conclusie	8
Steps in the Boot Process.....	9
Monitoring and Troubleshooting the Boot Process.....	11
System Recovery Techniques	12
Exam Essentials	13
Linux Boot Process:	13
Linux GRUB Legacy and GRUB2 Bootloaders:	14
Alternative Linux Bootloaders:	15
Recovering from a Kernel Panic:	15
Conclusion	16

Hoofdstuk 5 Uitleg de boot proces (NL)

Het Opstartproces van een Linux-systeem

Het opstartproces van een Linux-systeem bestaat uit een reeks gecoördineerde stappen die beginnen zodra de computer wordt ingeschakeld en eindigen wanneer het besturingssysteem volledig is opgestart en klaar voor gebruik. Dit proces omvat verschillende cruciale componenten zoals de firmware, bootloaders, de Linux-kernel en init-systemen. Het begrijpen van deze stappen is essentieel voor systeembeheerders en voor het oplossen van opstartproblemen.

Het Linux-bootproces kan worden opgesplitst in drie hoofdfasen:

1. De werkstationfirmware wordt gestart, voert een snelle controle van de hardware uit (dit wordt een Power-On Self-Test of POST genoemd) en zoekt vervolgens naar een opstartprogramma (bootloader) om uit te voeren vanaf een opstartbaar apparaat.
2. De bootloader wordt uitgevoerd en bepaalt welk Linux-kernelprogramma geladen moet worden.
3. Het kernelprogramma wordt in het geheugen geladen en start de noodzakelijke achtergrondprogramma's die vereist zijn voor de werking van het systeem (zoals een grafische desktopmanager voor desktops of web- en databaseservers voor servers).

Stappen in het Opstartproces

Firmware Start

- **BIOS (Basic Input/Output System):**
 - Voert de **Power-On Self-Test (POST)** uit om te controleren of de hardware van de computer correct werkt.
 - Zoekt naar een opstartbaar apparaat dat een **bootloader** bevat en laadt deze.
 - Beperkt tot 16-bits code en ondersteunt geen moderne schijven die groter zijn dan 2 TB.
- **UEFI (Unified Extensible Firmware Interface):**
 - Vervangt het oude BIOS en biedt een **grafische interface** en **netwerkcapaciteiten**.
 - Ondersteunt **secure boot**, wat betekent dat alleen ondertekende bootloaders geladen kunnen worden om de veiligheid te waarborgen.
 - Maakt gebruik van een **EFI System Partition (ESP)** om opstartbestanden en configuraties op te slaan.

Bootloader • LILO (Linux

Loader):

- Een ouder systeem dat direct gebruik maakt van schijfadressen en nu grotendeels verouderd is.
- Heeft geen ingebouwde **command-line interface** en ondersteunt geen complexe configuraties.
- **GRUB Legacy:** ○ Ondersteunt meerdere besturingssystemen en geavanceerde configuraties.
 - De configuratie wordt opgeslagen in `/boot/grub/menu.lst`. ○ Kan dynamisch worden aangepast via een command-line interface tijdens het opstarten.
- **GRUB2:**
 - Een veelgebruikte, flexibele bootloader die verbeterde functies biedt ten opzichte van zijn voorganger. ○ Configuraties worden opgeslagen in `/boot/grub/grub.cfg`, maar aanpassingen worden gedaan via `/etc/default/grub` en scripts in `/etc/grub.d`. ○ Ondersteunt **grafische menu's en theming**.

Kernel Load

- De geselecteerde **kernel** wordt in het geheugen geladen, samen met een **initramfs** (initial RAM filesystem) dat essentiële modules bevat voor het opstarten.
- De kernel initialiseert de hardwarecomponenten van de computer en start het initstelsel op.

Init-systeem

- **SysVinit:**
 - Een ouder init-systeem dat gebruik maakt van **runlevels** om verschillende sets van services te beheren.
 - De configuratiebestanden bevinden zich in `/etc/inittab`.
- **Systemd:**
 - Een moderne vervanger voor SysVinit, die gebruik maakt van **units** en **targets** in plaats van runlevels.
 - Biedt **parallele opstartmogelijkheden** en is verantwoordelijk voor het beheer van systemdiensten.
 - Configuratiebestanden worden verdeeld over `/etc/systemd/system/` en `/lib/systemd/system/`.

Bewaken en Probleemoplossing van het Opstartproces

Console Monitoring

Tijdens het opstartproces kunnen gebruikers de voortgang volgen via de **console-uitvoer**, die vaak nuttige informatie bevat over wat er tijdens het opstarten gebeurt.

- **Gebruik van dmesg:**
 - Het commando `dmesg` toont **kernel- en opstartberichten**, wat nuttig is voor het identificeren van hardware- of driverproblemen.
- **Logbestanden:**
 - Opstartberichten worden vaak opgeslagen in logbestanden zoals `/var/log/boot.log` of `/var/log/messages`, die gedetailleerde informatie bieden over het opstartproces.

Bootloader Configuratie en Herstel

- **GRUB2 Configuratie:**
 - Aanpassingen aan de GRUB2-configuratie worden gedaan in `/etc/default/grub`, waar je parameters zoals **GRUB_TIMEOUT** (tijd voor het menu) en **GRUB_CMDLINE_LINUX** (kernelparameters) kunt instellen. ◦ Na het aanpassen van de configuratie kun je `update-grub` gebruiken om de wijzigingen door te voeren en het configuratiebestand bij te werken.
 - **Herstel van Bootloaders:**
 - In geval van een beschadigde bootloader kan een **live Linux-omgeving** worden gebruikt om de bootloader opnieuw te installeren.
 - **GRUB2 Herstel:** Gebruik het commando `grub-install` gevolgd door `update-grub` om de bootloader opnieuw in te stellen.
-

Systeemherstel Technieken

Kernel Panics:

- Een **kernel panic** is een ernstige fout die optreedt wanneer de kernel niet correct kan functioneren, vaak door een probleem met hardware of de systeemconfiguratie.
- Kernel panics kunnen vaak worden opgelost door het opstarten in **single-user mode** of door herstelparameters toe te voegen via de bootloader.

Single-User Mode:

- **Single-user mode** wordt gebruikt voor onderhoudstaken zoals **wachtwoordherstel** of het oplossen van schijffouten.
- Het kan geactiveerd worden door "**single**" of "**rescue**" toe te voegen aan de kernelregel in GRUB.

Gebruik van Rescue Disks:

- In geval van ernstige systeemproblemen kan een **rescue disk** worden gebruikt om vanaf een alternatieve omgeving op te starten.
- Met deze disk kunnen systeemtools zoals **fsck** worden gebruikt om **bestandssysteemproblemen** te diagnosticeren en te repareren.

fsck Command:

- Het `fsck`-commando wordt gebruikt om fouten in bestandssystemen te controleren en te herstellen.
- Het wordt vaak uitgevoerd vanaf een live-omgeving om ervoor te zorgen dat het bestandssysteem niet in gebruik is tijdens de controle.

Examen Essentials

Linux Bootproces:

Het **Linux bootproces** kan in verschillende fasen worden onderverdeeld:

1. BIOS/UEFI Initialisatie:

- Wanneer je de computer inschakelt, is het **BIOS (Basic Input/Output System)** of **UEFI (Unified Extensible Firmware Interface)** verantwoordelijk voor het initialiseren van de hardware. Het voert een **Power-On Self-Test (POST)** uit om de systeemcomponenten zoals de CPU, het geheugen en de opslagapparaten te controleren.
- Zodra de hardware is geverifieerd, zoekt het BIOS/UEFI naar een **opstartbaar apparaat** (bijvoorbeeld een harde schijf, USB of CD/DVD) om de opstartlader (bootloader) van te laden.

2. **Bootloader Laden (GRUB):**

- Het BIOS/UEFI vindt en laadt een **bootloader**-programma van de **Master Boot Record (MBR)** (in het geval van BIOS) of de **EFI-systeempartitie** (voor UEFI). Veelgebruikte Linux-bootloaders zijn **GRUB Legacy** en **GRUB2**.
- De bootloader is verantwoordelijk voor het laden van de **Linux-kernel** in het geheugen, dit is het kernonderdeel van het Linux-besturingssysteem.

3. **Kernel Initialisatie:**

- Zodra de kernel is geladen, neemt de **Linux-kernel** het over, initialiseert hardwaredrivers en andere essentiële systeembewerkingen.
- De kernel zoekt naar het **init-programma** (of **systemd**, het moderne init-systeem) om systeemservices te starten.

4. **Systeemdiensten (init/systemd):**

- Het **init/systemd**-programma start verschillende essentiële achtergrondprogramma's, zoals netwerkdiensten, het mounten van bestanden en het voorbereiden van het systeem voor gebruik (bijvoorbeeld het starten van de **opdrachtregelterminals** of de **grafische desktopbeheerder**).

Linux GRUB Legacy en GRUB2 Bootloaders:

1. **GRUB Legacy:**

- **GRUB Legacy** (de oudere versie van GRUB) slaat de configuratiebestanden op in de map **/boot/grub**.
- Het gebruikt een configuratiebestand genaamd **menu.lst** of **grub.conf**, dat de opstartopties en de structuur van het menu definieert. Dit bestand kan commando's bevatten om een opstartmenu te maken waarmee je kunt kiezen tussen meerdere kernels, besturingssysteeminstallaties of opstartkenmerken.
- **Installatie:** Om GRUB Legacy te installeren, moet je het **grub-install**-commando gebruiken, waarmee de bootloader in de Master Boot Record (MBR) van de opstartschijf wordt geïnstalleerd.

2. **GRUB2:**

- **GRUB2** is de nieuwere versie van GRUB en slaat de bestanden ook op in de map **/boot/grub**, maar gebruikt een ander configuratiebestand: **grub.cfg**.
- In tegenstelling tot GRUB Legacy wordt **grub.cfg** niet direct bewerkt. In plaats daarvan worden configuratie-instellingen geplaatst in **/etc/default/grub** en aangepaste scripts worden opgeslagen in **/etc/grub.d/**.
- Het **grub-mkconfig**-commando genereert het **grub.cfg**-bestand door de configuraties uit de mappen **/etc/default/grub** en **/etc/grub.d/** te verzamelen. Zodra de configuratie is bijgewerkt, kun je deze installeren met het **update-grub**-commando.

Alternatieve Linux Bootloaders:

1. LILO (Linux LOader):

- **LILO** was een van de eerste Linux-bootloaders en wordt tegenwoordig niet veel meer gebruikt. Het gebruikt het configuratiebestand **/etc/lilo.conf** om de opstartopties en de te laden kernel te definiëren.
- Het mist enkele geavanceerde functies die in GRUB aanwezig zijn, zoals een opstartmenu en gemakkelijke kernel-updates.

2. SYSLINUX:

- **SYSLINUX** is een bootloader die is ontworpen voor **FAT-gebaseerde bestandssystemen**, zoals die op USB-stations en diskettes.
- Het wordt vaak gebruikt in lichte of embedded Linux-systemen.

3. ISOLINUX:

- **ISOLINUX** wordt gebruikt om Linux-systemen vanaf CD/DVD-media op te starten. Het is populair bij **LiveCD**-distributies.
- De bootloader wordt opgeslagen in het bestand **isolinux.bin** en de configuratieopties worden opgeslagen in het bestand **isolinux.cfg**.

4. PXELINUX:

- **PXELINUX** wordt gebruikt voor het opstarten van een systeem via het netwerk (PXE - Preboot eXecution Environment).
- Het bootloader-programma (**pxelinux.0**) wordt op de netwerkserver opgeslagen, samen met individuele configuratiebestanden die zich in de **pxelinux.cfg**-map bevinden. Elke werkstation kan een eigen configuratiebestand hebben.

5. EXTLINUX:

- **EXTLINUX** is een lichte bootloader die vaak wordt gebruikt in embedded Linux-systemen die een kleinere voetafdruk vereisen.

Herstellen van een Kernel Panic:

Als je systeem niet kan opstarten en een **kernel panic** tegenkomt, zijn er verschillende manieren om te herstellen:

1. Gebruik van het GRUB Opstartmenu:

- Bij het **GRUB opstartmenu** kun je de **E-toets** indrukken om de opstartmenu-invoer te bewerken.
- Dit stelt je in staat om extra kernelparameters toe te voegen, zoals het opstarten van het systeem in **single-user mode** of het gebruik van **safe mode**. Het toevoegen van de parameter **single** aan de kernel-opstartregel zal het systeem opstarten in de **single-user mode**, wat je mogelijk in staat stelt om problemen op te lossen.

2. Gebruik van een Herstel-CD:

- Als je systeem niet normaal kan opstarten, kun je een **herstel-CD** of **live Linux USB** gebruiken om in een **live-omgeving** op te starten.
- Eenmaal opgestart kun je hulpmiddelen zoals **fsck** gebruiken om corrupte bestandssystemen op de harde schijf te controleren en te repareren.
- Je kunt vervolgens het aangetaste bestandssysteem **mounten** met het **mount**-commando om systeembestanden te onderzoeken of te wijzigen die mogelijk de kernel panic veroorzaken.

Conclusie

Het **opstartproces van Linux** is een complex maar essentieel onderdeel van het systeembeheer. Het begrijpen van de verschillende fasen van het opstartproces en het vermogen om problemen in deze fase te diagnosticeren, helpt systeembeheerders om snel en effectief opstartproblemen op te lossen. Het beheersen van tools zoals **GRUB2** en het gebruik van systeemhersteltechnieken bieden de mogelijkheid om de meeste opstartproblemen te identificeren en te verhelpen.

Chapter 5 Explaining the boot process (EN)

The boot process of a Linux system consists of a series of coordinated steps that begin as soon as the computer is powered on and end when the operating system is fully loaded and ready for use. This process involves several critical components, including the firmware, bootloaders, the Linux kernel, and init systems. Understanding these steps is essential for system administrators and for troubleshooting boot issues.

The Linux boot process can be split into three main steps:

1. The workstation firmware starts, performing a quick check of the hardware (called a Power-On Self-Test, or POST) and then looks for a bootloader program to run from a bootable device.
2. The bootloader runs and determines what Linux kernel program to load.
3. The kernel program loads into memory and starts the necessary background programs required for the system to operate (such as a graphical desktop manager for desktops or web and database servers for servers).

Steps in the Boot Process

Firmware Start

- **BIOS (Basic Input/Output System):**
 - Performs the **Power-On Self-Test (POST)** to check if the hardware is functioning properly.
 - Searches for a bootable device containing a **bootloader** and loads it.
 - Is limited to 16-bit code and does not support modern drives larger than 2 TB.
- **UEFI (Unified Extensible Firmware Interface):**
 - A modern replacement for BIOS that offers a **graphical interface** and **networking capabilities**.
 - Supports **secure boot**, ensuring that only signed bootloaders can be loaded to ensure security.
 - Uses an **EFI System Partition (ESP)** to store boot-related files and configurations.

Bootloader

- **LILO (Linux Loader):**
 - An older system that directly uses disk addresses, now largely obsolete.
 - Does not have an integrated **command-line interface** or support for complex configurations.
- **GRUB Legacy:**
 - Supports multiple operating systems and advanced configurations.
 - Configuration is stored in `/boot/grub/menu.lst`.
 - Can be dynamically adjusted via a command-line interface during the boot process.
- **GRUB2:**
 - A widely used and flexible bootloader with enhanced features over its predecessor.
 - Configuration is stored in `/boot/grub/grub.cfg`, but adjustments are made through `/etc/default/grub` and scripts in `/etc/grub.d`.
 - Supports **graphical menus** and **theming**.

Kernel Load

- The selected **kernel** is loaded into memory, along with an **initramfs** (initial RAM filesystem) that contains essential modules for booting.
- The kernel initializes the system's hardware components and starts the init system.

Init System

- **SysVinit:**
 - An older init system that uses **runlevels** to manage different sets of services.
 - Configuration files are located in `/etc/inittab`.
 - **Systemd:**
 - A modern replacement for SysVinit that uses **units** and **targets** instead of runlevels.
 - Provides **parallel booting capabilities** and is responsible for managing system services.
 - Configuration files are spread across `/etc/systemd/system/` and `/lib/systemd/system/`.
-

Monitoring and Troubleshooting the Boot Process

Console Monitoring

During the boot process, users can monitor progress through **console output**, which often provides useful information about what is happening during startup.

- **Using dmesg:**
 - The `dmesg` command shows **kernel and boot messages**, which are helpful for identifying hardware or driver issues.
- **Log Files:**
 - Boot messages are often stored in log files such as `/var/log/boot.log` or `/var/log/messages`, which provide detailed information about the boot process.

Bootloader Configuration and Recovery •

GRUB2 Configuration:

- Changes to the GRUB2 configuration are made in `/etc/default/grub`, where parameters such as **GRUB_TIMEOUT** (time for menu) and **GRUB_CMDLINE_LINUX** (kernel parameters) can be set. ◦ After modifying the configuration, use `update-grub` to apply the changes and update the configuration file.
 - **Bootloader Recovery:**
 - In case of a damaged bootloader, a **live Linux environment** can be used to reinstall the bootloader.
 - **GRUB2 Recovery:** Use the `grub-install` command followed by `updategrub` to reinstall and configure the bootloader.
-

System Recovery Techniques

Kernel Panics:

- A **kernel panic** is a critical error that occurs when the kernel cannot function properly, often due to hardware issues or system misconfigurations.
- Kernel panics are often resolved by booting into **single-user mode** or adding recovery parameters through the bootloader.

Single-User Mode:

- **Single-user mode** is used for maintenance tasks such as **password recovery** or fixing disk errors.
- It can be activated by adding "**single**" or "**rescue**" to the kernel line in GRUB.

Using Rescue Disks:

- In case of severe system issues, a **rescue disk** can be used to boot from an alternative environment.
- With this disk, tools like **fsck** can be used to diagnose and repair **filesystem issues**.

fsck Command: • The `fsck` command is used to check and repair errors in filesystems.

- It is often run from a live environment to ensure that the filesystem is not in use during the check.

Linux Boot Process:

The **Linux boot process** can be broken down into several stages:

1. **BIOS/UEFI Initialization:**

- When you power on your computer, the **BIOS (Basic Input/Output System)** or **UEFI (Unified Extensible Firmware Interface)** is responsible for initializing the hardware. It performs a Power-On Self-Test (POST) to check the system components like the CPU, memory, and storage devices.
- Once the hardware is verified, BIOS/UEFI looks for a **bootable device** (e.g., hard drive, USB, or CD/DVD) to load the bootloader from.

2. **Bootloader Loading (GRUB):**

- The BIOS/UEFI finds and loads a **bootloader** program from the **Master Boot Record (MBR)** (in the case of BIOS) or the **EFI system partition** (for UEFI). Common Linux bootloaders include **GRUB Legacy** and **GRUB2**.
- The bootloader is responsible for loading the Linux kernel into memory, which is the core part of the Linux operating system.

3. **Kernel Initialization:**

- Once loaded, the **Linux kernel** takes over, initializing hardware drivers and other essential system processes.
- The kernel looks for the **init program** (or **systemd**, the modern init system) to start system services.

4. **System Services (init/systemd):**

- The init/systemd program starts various essential background programs, such as **network services**, **mounting filesystems**, and preparing the system for use (e.g., starting the **command-line terminals** or **graphical desktop manager**).
-

Linux GRUB Legacy and GRUB2 Bootloaders:

1. GRUB Legacy:

- **GRUB Legacy** (the older version of GRUB) stores its configuration files in the `/boot/grub` directory.
- It uses a configuration file called **menu.lst** or **grub.conf**, which defines the boot options and menu structure. This file can include commands to create a boot menu for selecting between multiple kernels, OS installations, or boot features.
- **Installation:** To install GRUB Legacy, you must run the `grub-install` command, which installs the bootloader into the Master Boot Record (MBR) of the boot disk.

2. GRUB2:

- **GRUB2** is the newer version and stores its files in the `/boot/grub` directory as well, but it uses a different configuration file: **grub.cfg**.
 - Unlike GRUB Legacy, **grub.cfg** is not edited directly. Instead, configuration settings are placed in `/etc/default/grub` and custom scripts are placed in `/etc/grub.d/`.
 - The `grub-mkconfig` command generates the `grub.cfg` file by gathering the configurations from the `/etc/default/grub` and `/etc/grub.d/` directory. Once the configuration is updated, you can install it using the `update-grub` command.
-

Alternative Linux Bootloaders:

1. LILO (Linux LOader):

- **LILO** was one of the earlier Linux bootloaders and is no longer commonly used. It uses the configuration file `/etc/lilo.conf` to define the boot options and which kernel to load.
- It lacks some of the advanced features present in GRUB, such as a boot menu and easy kernel updates.

2. SYSLINUX:

- **SYSLINUX** is a bootloader designed for FAT-based filesystems like those on USB drives and floppy disks.
- It's commonly used in lightweight or embedded Linux systems.

3. ISOLINUX:

- **ISOLINUX** is used to boot Linux systems from CD/DVD media. It's popular with **LiveCD** distributions.
- The bootloader is stored in the `isolinux.bin` file, and configuration options are stored in the `isolinux.cfg` file.

4. PXELINUX:

- **PXELINUX** is used for booting a system over the network (PXE - Preboot eXecution Environment).
- The bootloader program (`pxelinux.0`) is stored on the network server, along with individual configuration files located in the `pxelinux.cfg` directory. Each workstation can have its own configuration file.

5. EXTLINUX:

- **EXTLINUX** is a lightweight bootloader often used in embedded Linux systems that require a smaller footprint.

Recovering from a Kernel Panic:

If your system fails to boot and encounters a **kernel panic**, there are several ways to recover:

1. Using GRUB Boot Menu:

- At the **GRUB boot menu**, you can press the **E** key to edit the boot menu entry.
- This allows you to add kernel parameters, such as starting the system in **single-user mode** or using **safe mode**. For example, adding `single` to the kernel boot line will boot the system into single-user mode, which might allow you to fix issues.

2. Using a Rescue Disk:

- If your system cannot boot normally, you can use a **rescue disk** or live Linux USB to boot into a **live environment**.
- Once booted, you can use tools like `fsck` to check and repair corrupted filesystems on the hard drive.
- You can then mount the affected filesystem using the `mount` command to examine or modify system files that might be causing the kernel panic.

Conclusion

The **boot process of Linux** is a complex but essential part of system administration. Understanding the different phases of the boot process and being able to diagnose issues at this stage helps administrators quickly and effectively resolve boot-related problems. Mastering tools like **GRUB2** and using system recovery techniques provides the capability to address and fix most boot problems.