

## 101: Hardware en architectuur [8]

### Boek 2, hoofdstuk 1. Bladzijde 1 t/m 26

101.1 Het configureren van fundamentele systeemhardware BIOS (1)

101.2 vervallen

101.3 Het configureren van modems en geluidskaarten (1)

101.4 Het configureren van SCSI devices (1)

101.5 Het configureren van diverse pc-uitbreidingskaarten (3)

101.6 Het configureren van communicatie devices (1)

101.7 configureren USB devices [1] handleiding blz 16

#### 101.1 BIOS (Basic Input Output System)

Als de computer aan wordt gezet is het helemaal leeg, een van de eerste zaken die er dan gebeuren is dat de processor een resetsignaal krijgt. Na dit signaal gaat de processor een programma uitvoeren dat op een vaste plaats in het geheugen staat. Omdat dit programma in het **ROM geheugen** staat is het programma altijd aanwezig. Dit programma wordt het **bios** genoemd en het handelt de meest eenvoudige bewerkingen af: een test voor het beschikbare geheugen, verschillende soorten harddisks, instellen datum en tijd. Datum en tijd staat niet in het ROM, die gegevens moeten door RAM geheugen worden aangeleverd. Vandaar dat iedere PC een klein stukje **RAM** heeft, met aparte voedingsspanning (kleine batterij): het **BIOS-geheugen**. Het aanpassen van een aan instellingen in het BIOS kan door tijdens het booten de F2-toets in te drukken. U komt dan in de BIOS-setup. Hier kunnen en aantal gegevens worden gewijzigd: tijd, opstarten/booten van welke devices. Een van de instellingen in het BIOS betreft de aanwezige IDE harddisks. Variabelen daarbij zijn: het aantal cilinders, het aantal koppen en het aantal sectoren per track. Als het aantal waarden van C x H x S X 512 bytes niet overeenkomen met de eigenschappen van de harddisk zal het systeem niet kunnen opstarten. Linux gebruikt maar heel weinig gegevens vanuit het BIOS, dit in tegenstellingen tot andere besturingssystemen. Maar eerst moet Linux wel vanuit het BIOS worden gestart om daarna direct naar de kernel door te gaan. Daarom moeten deze aanwijzingen zich al in het begin van de harddisk bevinden. Exact: onder de 1024ste cilinder. Hoewel: met GRUB is er meer mogelijk. Het beste is om altijd in het BIOS het gebruik van **LBA** of large instellingen te vermijden. **LBA** [large block addressing] is als oplossing gekomen om het cilindersprobleem te omzeilen. Door het aantal cilinders in het bios te halveren en de lees- en schrijfkoppen te vergroten kon men toch alle blokken op een harddisk adresseren. Helaas zijn ze voor andere besturingssystemen wel noodzakelijk, dus eenmaal ingesteld kunt u ze niet meer veranderen.

**Het proc-systeem:** geen onderdeel van de harddisk, maar een directory die direct toegang geeft tot de kernel. Binnen deze directory-structuur bestaan veel bestanden om de werking van de kernel mee te beïnvloeden of juist uit te lezen.

**Interrupts:** In de PC zitten allerlei onderdelen die op een complexe wijze met de kernel communiceren. Als zo'n onderdeel de processor wil laten weten dat er iets gedaan moet worden genereert zo'n onderdeel een bepaalde interrupt. Omdat er meerdere onderdelen in een PC zitten zijn er ook meerdere interrupts nodig. Ieder onderdeel heeft zijn eigen interrupt of IRQ nummer tussen de 0 en 16. De oudere ISA kaarten moeten vaak handmatig een IRQ nummer ingesteld krijgen. De moderne PCI kaarten overleggen zelf met het systeem welk IRQ zij zullen krijgen.

**BASE IO.** Naast het interrupt gegeven is het ook mogelijk voor onderdelen om soms een interrupt te delen. Door hetzelfde IRQ nummer te gebruiken maar verschillende Base IO adressen kunnen zij toch onafhankelijk van elkaar een interrupt genereren. Sommige kaarten gebruiken ook nog DMA en MEM adressen. Deze worden gebruikt voor het snelle transport van grote hoeveelheden data.

**Plug and Play.** Een modernisering van de ISA kaarten leverde het plug and Play (pnp) mechanisme op. Bij linux kan handmatig met het **pnpdump** commando een overzicht worden verkregen van wie welke interrupt gebruikt. U kunt dan instellingen lezen en met **isapnp** toewijzen.

**Standaardinstellingen.** Op een PC is er altijd een aantal standaard-instellingen oor zowel de IRQ als de Base IO waarden. Doorgaans 15. Met hetbestandstype **/proc** kunt u waarden inlezen en toewijzen. Dit heeft dus direct consequenties voor de kernel.

**cat /proc/ interrupts** geeft de **IRQ** bezetting | **cat /proc/ioports** geeft de **BASE IO**.

```
linux:/home/rene # cat /proc/interrupts
CPU0
0: 12699119      XT-PIC timer
1: 25226         XT-PIC i8042
2: 0            XT-PIC cascade
5: 84474        XT-PIC ALI 5451
9: 2986         XT-PIC acpi
10: 66429       XT-PIC ohci1394, uhci_hcd, uhci_hcd, eth0
11: 6           XT-PIC yenta, ehci_hcd
12: 1916412     XT-PIC i8042
14: 122661      XT-PIC ide0
15: 93171       XT-PIC ide1
NMI: 0
LOC: 0
ERR: 1
MIS: 0
```

```
linux:/home/rene # cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
.... etc
0376-0376 : ide1
0378-037a : parport0
2040-2047 : ide0
2048-204f : ide1
2400-24ff : 0000:00:12.0
etc.
```

Op een standaard PC zijn er altijd een aantal gebruikelijke instellingen voor zowel de IRQ als de BASE IO waarden:

Onderdeel	irq	base io
Pc timer	0	0040-005f
toetsenbord	1	0060-006f
cascade	2	
2de seriele poort	3	02f8-02ff
1ste seriele poort	4	03f8-03ff
4de seriele poort	3	02e8-02ef
3de seriele poort	4	03e8-03ef
2de parallele poort	5	
diskette controller	6	
1ste parallele poort	7	0378-037f
real time clock [rtc]	8	0070-007f
PS/2 muis	12	
Floating point Unit	13	00F0-00FF
1STE ide controller	14	01f0-01f7 en e000-e007 en 03f6
2de ide controllers	15	0170-0177 en e008-e00f en 0376

Zoals u ziet blijven er niet veel IRQ's over voor uitbreidingen van het systeem. De IRQ's 9, 10 en 11 zijn vaak beschikbaar. Dit maakt het configureren van de hardware er niet eenvoudiger op.

**101.4 SCSI.** De afkorting SCSI staat voor Small Computer System Interface. Het is een systeem waarbij er in de computer een uitbreidingskaart wordt gemonteerd. Dit is de SCSI controller. Vervolgens kunnen er allerlei SCSI apparaten aan deze controller worden gekoppeld. Dit gebeurt door middel van een lange kabel met daarop een groot aantal aansluitingen voor SCSI apparaten. De meest voorkomende typen apparaten zijn harddisks, CD-ROM spelers en scanners. Doordat al deze apparaten de aansluitkabel met elkaar delen dienen ze op een of andere manier een unieke code te krijgen. De **SCSI** heeft ook een **eigen ID**, veelal is dit ID 7. De SCSI controller heeft vaak software aan boord (**SCSI BIOS**) voor de communicatie met de apparaten. Een van de taken is van het SCSI BIOS is het uitvoeren van een scan op de SCSI BUS naar de aangesloten apparaten. Dit gebeurt bij het opstarten van het systeem en vaak meldt het systeem ook welke apparaten (devices) hij heeft geïdentificeerd. Als er vanaf een **SCSI harddisk** moet worden **geboot** dan dient deze harddisk altijd het **SCSI ID 0** (nul) te hebben. Het SCSI systeem is een voorbeeld van een **bus-systeem**.

Op deze bus sluit men apparaten aan en door hun onderlinge nummer kan er in de communicatie geen verwarring ontstaan. Een van de zaken waar men bij een bus-systeem op moet letten is het afsluiten van de kabel. Elke SCSI kabel, ongeacht hoeveel apparaten er op zijn aangesloten, moet aan beide zijden zijn **getermineerd**. Soms is dit al gemonteerd in de computer.

**SCSI** heeft een mapbestand in `/proc/scsi`. Daarin staan [SUSE] ook **device\_info** [informatie over drivers] en het bestand **scsi** [welke apparaten er aangesloten zijn]. [In het boek spreken ze over **scsi\_info, proc/scsi en SCSI ID!!**]

**101.3 Modems.** Een modem is een apparaat waarmee een computer over een telefoonlijn een verbinding kan opzetten met een andere computer. Van oudsher kent men twee typen modems: het interne en het externe model. Het interne model is een uitbreidings-kaart voor in de PC en het externe model is een kastje dat aangesloten wordt op de seriële poort van de computer. Er zijn ook zogenaamde winmodems, hierbij is om op de hardware van het modem te besparen en een aantal rekentaken door de processor van de PC te laten doen. Dat werkt prima bij besturingssystemen die single-user zijn, maar het is een minder geslaagd idee voor Linux met zijn multi-user architectuur. Houd er rekening mee dat deze modems niet of nauwelijks op een linux systeem kunnen worden aangesloten. Voor **modems** geldt ook weer dat er **unieke IRQ's, IO adressen** dienen te worden gebruikt. Dat kan vaak handmatig op de modemkaart worden ingesteld of via software als het een PNP modem is (**isapnp, pnpdump**). **Externe modems** kunnen natuurlijk geen IRQ's toegewezen krijgen omdat zij zijn aangesloten op een **seriële poort** met zijn eigen IRQ-instelling.

**bitrate.** seriële verbindingen zijn verbindingen waarbij de gegevens achter elkaar (in serie) worden verstuurd. Dit in tegenstelling tot parallelle verbindingen waarbij gegevens naast elkaar (parallel) worden verstuurd. Bij serie verbindingen is het belangrijk dat zowel de zender als de ontvanger dezelfde snelheid hanteren. (56000k-115200k).

**setserial** wordt gebruikt voor het instellen van de seriële poort. Dit commando wordt gebruikt voor het afstemmen van de instellingen in de hardware.

```
linux:/home/rene # setserial -gva /dev/ttyS0
/dev/ttyS0, Line 0, UART: 8250, Port: 0x1440, IRQ: 10
  Baud_base: 115200, close_delay: 500, divisor: 0
  closing_wait: 30000
  Flags: spd_normal skip_test
```

Op deze manier kunt u de huidige instellingen achterhalen. Hieruit blijkt dat de IRQ 10 is en de IO 0x1440. De eigenlijk snelheid van de verbinding staat er niet, wel de hoogste mogelijke snelheid (baud\_base 115200). Met behulp van **stty** 115200 (bv) kan een snelheid worden opgegeven (alleen die snelheid die binnen het normale spectrum valt: 33600-9600-57600-115200).

Samenvattend is de beste volgorde voor het werken met modems: instellen van de hardware via jumpers of het **isapnp** commando, vervolgens met **setserial** een hardware poort toewijzen aan een software poort en als laatste de communicatie-snelheid instellen met het **stty** commando.

**Modemconfiguratie.** De eerste populaire modems werden geproduceerd door een bedrijf genaamd 'Hayes'. Een van de ontwikkelingen van dit bedrijf was een modem-communicatie-taal. Met deze taal konden allerlei acties op een modem worden verzorgd. Elk commando van het modem begint met het AT. Daarna komt de gewenste actie of instelling. Als een commando wordt begrepen en is uitgevoerd wordt de tekst OKE teruggestuurd. De belangrijkste commando's:

```
ATDT0,072. Bel met toonsignaal (Dial Tone) een nul wacht dan even en bel daarna met 072
ATH0. Hang de verbinding op
ATSO=2. Neem na twee belsignalen de telefoon op voor een inkomende verbinding
ATW. Het vastleggen van de gewijzigde instellingen in het modem
ATZ. Reset het modem. Vergeet alle wijzigingen en herstart met de opgeslagen instellingen
AT&H. Geeft doorgaans de helpetekst weer.
```

**Geluidskaarten.** Voor de ondersteuning van geluidskaarten onder linux geldt hetzelfde als voor andere uitbreidingskaarten. Zorg ervoor dat er unieke IRQ's e.d. worden gebruikt. Linux ondersteunt een groot aantal verschillende geluidskaarten. Veelal is de ondersteuning afhankelijk van de geluidschips op de kaarten. Het is dus belangrijk om typenummers te noteren en te vergelijken met de lijst van ondersteunende chips. Er zijn drie bronnen voor drivers voor linux: in de kernel zelf, het Open Sound systeem en het ALSA project. Het boek spreekt nog over **sndconfig** [alleen geldig voor Red Hat?]

**101.5 PCI.** De PCI local bus is een type slot voor PCI uitbreidingskaarten. Het grote voordeel van PCI kaarten is dat zij in staat zijn om zichzelf te configureren (vergelijkbaar met Plug and Play van ISA, maar dan veel beter). **lspci** geeft informatie over de PCI-kaarten. Optie **-v** geeft meer informatie en de optie **-vv** geeft alle informatie die de kaart kan mededelen.

Met **setpci** kunt u de devices zowel ondervragen als instellen. Zo kunt u bepaalde waarden in bepaalde registers van bepaalde PCI kaarten plaatsen. Het moge duidelijk zijn dat dit niet normaal werk voor een startende linux gebruiker is.

```
linux:~ # lspci
0000:00:00.0 Host bridge: ATI Technologies Inc RS200/RS200M AGP Bridge [IGP 340M] (rev 02)
0000:00:01.0 PCI bridge: ATI Technologies Inc PCI Bridge [IGP 340M]
0000:00:06.0 Multimedia audio controller: ALi Corporation M5451 PCI AC-Link Controller Audio Device (rev 02)
0000:00:07.0 ISA bridge: ALi Corporation M1533 PCI to ISA Bridge [Aladdin IV]
0000:00:08.0 Modem: ALi Corporation M5457 AC'97 Modem Controller
0000:00:0a.0 CardBus bridge: O2 Micro, Inc. OZ6912 Cardbus Controller
0000:00:0b.0 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1 Controller (rev 50)
0000:00:0b.1 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1 Controller (rev 50)
0000:00:0b.2 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 51)
0000:00:0c.0 FireWire (IEEE 1394): Texas Instruments TSB43AB21 IEEE-1394a-2000 Controller (PHY/Link)
0000:00:10.0 IDE interface: ALi Corporation M5229 IDE (rev c4)
0000:00:11.0 Bridge: ALi Corporation M7101 Power Management Controller [PMU]
0000:00:12.0 Ethernet controller: National Semiconductor Corporation DP83815 (MacPhyter) Ethernet Controller
0000:01:05.0 VGA compatible controller: ATI Technologies Inc Radeon IGP 340M
```

**101.7 USB** {universal serial bus} is een plug an play interface tussen computer en externe devices [zoals audiospelers, scanners en printers]. Met usb kan een nieuwe device aan de computer worden gekoppeld zonder speciale adapterkaart. Het is een volledig open technologie waarop iedereen mag uitbreiden. Een usb aansluiting is tevens in staat om kleine apparaten van stroom te voorzien. USB kunnen vaak onderling met elkaar woren verbonden. Linux ondersteunt ook usb, maar men moet zich realiseren dat usb slecht een communicatiestandaard is.

Zo zijn er drivers/modules die het algemene deel voor hun rekening nemen [**usbcore**], modules die specifieke chipsets ondersteunen [**ihci**, **ohci**] en modules die algemene usb functionaliteit verzorgen [**usb storage**]. Er dient een usb ondersteuning in de kernel te zijn [vanaf 2.2.7]. Daarna is de driver voor de generieke usb-ondersteuning nodig: **usbcore.o** en daarna de usb chipsetcontroller in uw systeem.

Met **lspci** kunt u kijken welk type controller uw computer heeft: UHCI: Intel en Via moederborden of OHCI: ali en Sis chipsets. Hierna wordt de algemene functionaliteit module geladen [mouse, scanner, opslagmedia]. Er zijn twee pakketten die het aansluiten van usb-apparaten makkelijker maken: **usbmgr** en **hotplug**.

**usbmgr.** Dit programma moet draaien en met behulp van **dump\_usb/dev** kan een uitdraai worden gemaakt van de aangesloten usb apparaten. Dit kan vervolgens in als een bestand worden opgeslagen in **/etc/usbmgr/usbmgr.conf**

**hotplug.** Een wat nieuwere ontwikkeling is hotplug. Hiermee beoogt de software om de configuratie automatisch te laten plaatsvinden. Niet alleen voor USB en PCI, maar ook voor PCMCIA, Firewire. De nodige configuratiebestanden staan in **/etc/hotplug**

## 102: Devices, partities en filesystemen (22)

Boek 2, hoofdstuk 2. Bladzijde 37 t/m47

102.1 Het ontwerpen van een harddisk-layout (5)

102.2 Het installeren van een boot-manager + GRUB (1) handleiding bladzijde 23

Boek 2, hoofdstuk 3. Bladzijde 49 t/m 69

102.3 Het compileren en installeren van programma's van broncode (5)

102.4 Het beheer van gedeelde bibliotheken (3)

102.5

102.6 Het gebruik van RedHat Package Management (8) + handleiding bladzijde 30

**Primaire en logische partities.** Van de laatste kunt u er in theorie een onbeperkt aantal hebben. In de praktijk is dat voor SCSI-disks maximaal 15 en voor IDE-disks 63. De nummering van devicefiles voor IDE disks start vanaf hda. Dus de eerste partitie op de eerste IDE disk is /dev/hda1. De nummering voor SCSI disks start vanaf sda.

**Benodigde partities.** Als u meerdere besturingssystemen vanaf dezelfde disk wilt opstarten, dan moeten alle opstart-partities allemaal onder de 1024ste cilinder te komen. Als u dus zowel linux als windows NT en OS2 wilt kunnen opstarten dient u drie primaire partities te definiëren, alle onder de 1024ste cilinder. Als u alleen een linux wilt opstarten maakt u een partitie die begint en eindigt voor de 1024ste cilinder. Naast de boot-partitie die in ieder geval de kernel bevat dient u minimaal een partitie te maken voor de rest van het besturings-systeem en gegevens. Daarnaast moet u nog een of meerdere swap-partities maken.

**Swap-partitie** is een daarvoor gecreëerde ruimte op de harddisk en is een mechanisme waarmee het interne geheugen (het RAM) van een computer kan worden uitgebreid. Door een deel te reserveren als swap-space komt er meer geheugen ter beschikking van de kernel. Zodra er meer behoefte bestaat aan geheugen dan er feitelijk te beschikking is wordt een deel van het geheugen naar swap-space verplaatst (vandaar ook de term swap wat wisselen betekent) Tegnwoordig worden alleen die delen naar de swap-space verplaatst in blokken van 4kb tegelijk. Swap-space kan op twee manieren worden verkregen. De eerste is een aparte swap-partitie. Deze wordt alleen gebruikt door de kernel en deze omzeilt het bios. Dit betekent dat de swap-space overal op de harddisk mag staan. Een swap-partitie wordt gemaakt door **mkswap** + deel van de disk /dev/hdX. Met **swapon** dev/hdX wordt de wisselruimte geactiveerd. Een oude regel is dat meer dan 3 maal het RAM-geheugen aan swap-ruimte teveel is: het wordt in de praktijk dan nooit gebruikt. Houd 2 keer aan: 512 megabyte RAM = 1024 megabyte SWAP. Zie hieronder dat hda1 een swappartitie is. Het commando fdisk komt later aan de orde. Opvallend is ook het aantal cilinders, GRUB heeft daar echter minder moeite mee.

```
linux:/home/rene # fdisk /dev/hda
```

Het aantal cylinders van deze schijf is ingesteld op 77520.

Hier is niets mis mee, maar het is groter dan 1024, en kan met bepaalde instellingen problemen veroorzaken met:

- 1) software die werkt bij het opstarten (bijv. oude versies van LILO)
- 2) opstart- en partitioneringssoftware van andere besturings-systemen, zoals DOS FDISK en OS/2 FDISK

Opdracht (m voor hulp): p

```
Schijf /dev/hda: 40.0 GB, 40007761920 bytes
16 koppen, 63 sectoren/spoor, 77520 cylinders
Eenheden = cylinders van 1008 * 512 = 516096 bytes
```

Apparaat	Boot	Start	Einde	Blokken	Id	Systeem
/dev/hda1		1	756	380992+	82	Linux-swap / Solaris
/dev/hda2	*	757	77520	38689056	83	Linux

**Partitie-grootte.** Bij het partitioneren van een harddisk gelden regels omtrent de beste grootte van een partitie. Dit is natuurlijk afhankelijk van het gebruik van het systeem. Het **root** filesystem zelf hoeft in principe niet groot te zijn. Als u uw systeem voorziet slechts een filesystem dan moet dit natuurlijk groot genoeg zijn om alle bestanden te bevatten. Maar deze indeling is niet aan te

raden. Als u **/usr** op een aparte partitie zet en u bent van plan om veel software te gaan installeren dan is 9 GB tegenwoordig bijna een minimum. In **/opt** kunt u nog additionele pakketten installeren. KDE en GNOME bevinden zich meestal in deze directory. Een **/opt** van 1 GB is dan zeker geen overbodige luxe. Als u uw systeem gaat gebruiken als USENET (=news) dan dient u veel ruimte te maken in het filesysteem waar **/var/spool/news** in gaat vallen. Als u veel news wilt opslaan moet u al snel aan minimaal 5 GB denken (of 20 GB als u echt alles wilt).

**superblock.** In Linux, partitions are discussed in terms of blocks. A block is 512 bytes. The superblock is the first 512 bytes of a partition. An fs-specific data structure, stored on disk, that holds various bits of information about the fs on the disk, such as a pointer to the free blocks structure, the number of free inodes, etc. In the code, the inode manipulation routines are entered in super\_operations vector of function pointers. Het **superblock** wordt o.a gebruikt voor het controleren van het filesysteem [reiserfsck].

**Bootmanager LILO.** Een klein stukje software dat wordt gebruikt om het besturingssysteem te starten. Als LILO het linux systeem opstart (boot) dan gebruikt het LILO programma BIOS routines om de kernel te laden. Omdat er op dat moment nog geen directory-structuur aanwezig is worden de exacte diskblok-nummers gebruikt. Deze worden tijdens de installatie van LILO dan ook ingesteld. Omdat het BIOS vaak moeite heeft met disk-cilinders boven de 1023 moet de kernel dan ook onder de 1024ste cilinder zijn geplaatst. Dat betekent ook dat u elke keer als u de boot-opties of kernel wijzigt LILO zelf opnieuw moet installeren. Er zijn een aantal plaatsen waar u **LILO** software kunt **installeren:**

Op **diskette:** een snelle en veilige manier om op uw systeem met meerdere besturingssystemen om te gaan. Als de diskette niet in de drive kunt u een ander besturingssysteem laten starten. Wel dient in het BIOS te staan ingesteld dat de diskette tijdens het opstarten gecontroleerd wordt (en dus voor de harde schijf wordt gecontroleerd) In het **MASTER BOOT RECORD** (MBR) van de eerste harddisk. Voor IDE is dat **/dev/hda** en voor SCSI drive **/dev/sda**. Het MBR is het deel van de harddisk dat als eerste gecontroleerd wordt. In de **boot-sector**. Op de eerste sector van de bootpartitie. Die partitie dient dan wel als active te worden/zijn ingesteld.

Als **LILO** is geïnstalleerd als **bootmanager** kan men deze op de volgende manier gebruiken. Tijdens het opstarten verschijnt als alles goed gaat de tekst LILO. Vervolgens kunt u via tab een lijst krijgen met mogelijkheden. Enkele daarvan zijn:

```
root= om een andere rootpartie mee aan te geven
init= kan gebruikt worden om deze te definiëren. U kunt dan een runlevel opgeven, en bij problemen init-bin/sh. U krijgt dan een shell om mogelijke problemen op te lossen.
```

De configuratie van LILO gebeurt middels het **/etc/lilo.conf** bestand. Daarin staan alle instellingen. Het LILO.conf bestand bestaat uit een algemeen deel met instellingen en daarna een of meerdere sets met instellingen van de kernel.

```
algemeen:
boot = /dev/hda # of u eigen rootpartie
delay = 10 # wacht zoveel tienden seconden op interactie
vga = normaal # optioneel.
#linair = # gebruik deze optie bij disk-geometrie problemen.
delay 50 # wacht 5 seconden en start anders het eerste kernel-deel
```

```
Kernel-deel
image = boot/vmlinux # de kernel
root = /dev/hda1 # de root partitie
label = linux # naam waarin u de instellingen kunt herkennen
read-only # het root filesysteem read-only mounten
```

Het rootfile-systeem zal eerst read-only worden gemount en vervolgens via een opstartscript in een later stadium de partitie read-write mounten. U kunt aan LILO ook een ander besturingssysteem toevoegen. Bijvoorbeeld:

```
other = dev/hda4 # bv dos partitie
label = dos
```

## GRUB

De nieuwe bootloader is **GRUB** [grand unified bootloader]. Hier speelt de 1024 cilinder grens niet omdat er minder informatie van het BIOS wordt gebruikt. Grub zal LILO van de troon stoten. Voordelen : in overeenstemming met multiboot specificatie, basisfuncties zijn eenvoudig in te stellen doch veel mogelijkheden voor experts...enz. Grub gebruikt een configuratiebestand [menu.lst of grub.conf] dat rechtstreeks door de bootloader wordt gelezen. Grub telt disks vanaf 0. Dus mijn linux /dev/hda2 is bij grub hd[0,1].

Grub kent drie stadia. Stadium 1 is de **first stage bootloader**, die in de eerste sector wordt geplaatst. Vervolgens wordt 1.5 geplaatst. Als laatste wordt 2.0 gestart. Met grubinstall kunt u grub op de harddisk plaatsen. Wilt u grub niet in het MBR maar in de 1ste sector van de partitie, dan is het commando niet setup[hd0] maar setup[hd0,0]. Hoe alles binnen grub wordt opgestart wordt in het **/boot/grub/.menu.lst** geregeld [**SUSE**] of **boot/grub/grub.conf**

```
linux:# cat /boot/grub.menu.lst

# Modified by YaST2. Last modification on di apr 26 20:13:11 CEST 2005

color white/blue black/light-gray
default 0
timeout 1
gfxmenu (hd0,1)/boot/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title SUSE LINUX 9.3
    kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 vga=0x317 selinux=0
splash=silent resume=/dev/hda1 showopts
    initrd (hd0,1)/boot/initlinux:/boot/grub # cat menu.lst

###Don't change this comment - YaST2 identifier: Original name: floppy###
title Diskette
    root (fd0)
    chainloader +1

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe -- SUSE LINUX 9.3
    kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 showopts ide=nodma apm=off
acpi=off vga=normal noresume selinux=0 barrier=off nosmp noapic maxcpus=0 3
    initrd (hd0,1)/boot/initrd
```

Nadat u het systeem opnieuw heeft opgestart, wordt een menu gepresenteerd waaruit u de gewenste systeemomgeving/kernel kunt kiezen. U kunt echter ook op de toets ESC drukken waarmee u een grubshell gepresenteerd krijgt. Hiermee kunt u diverse grub- commando's geven of individuele menu-entries aanpassen.

### 102.3 Software installeren vanaf broncode.

Veel van de linux software wordt in broncode verspreid. Broncode is de vorm waarin de software door programmeurs wordt geschreven. Na het schrijven dient deze software vertaald te worden naar een voor de computer begrijpelijke vorm. Deze vertaalslag wordt compileren genoemd. De software wordt vaak in vorm van gecomprimeerd tar-archief gedistribueerd. Op een linux systeem is de directory **usr/src** gereserveerd voor software die in broncode wordt verspreid. In deze map is ook vaak nog een directory linux te vinden waar u de broncode van de kernel kunt vinden.

**Uitpakken.** De bestanden worden vaak als **bestandsnaam.tar.gz** of **bestandsnaam.tgz** verspreid. **Gunzip** is een compressieprogramma dat op elk linux-systeem te vinden is. Het commando voor **comprimeren** is **gzip** en voor **decomprimeren** **gunzip** of (**gzip -d**). Als u dus een bestand met de naam **bestandsnaam.tar** krijgt moet u deze optie weglaten. **Bzip2:** werkt nagenoeg gelijk aan gunzip, inclusief uitpak-opties

Bij het bekijken van de inhoud is het belangrijk om te constateren dat alle bestanden beginnen met een directorynaam. Dat betekent dat deze directory bij het uitpakken zal worden aangemaakt. Als alles goed is kan daarna het gecomprimeerde bestand worden uitgepakt met behulp van **tar.bestandsnaam -xzvf.**

**x=** extract : pakt uit  
**z=** comprimeert of decomprimeert de gegevens tijdens verwerking  
**v=** verbose: als u wilt zien wanneer er wat gebeurt  
**f=** file. Om de file op te geven

Dit uitpakken gebeurt in een directory van `/usr/src`. Als alles goed is kunt in die directory een aantal bestanden zien:

**README:** Is vaak het eerste bestand dat u moet lezen. Het bevat allerlei nutige informatie over de software en verdere verwijzingen naar andere bestanden

**INSTALL:** bevat informatie hoe de software geïnstalleerd dient te worden. Hier gaan we uit van de standaard-manier van GNU software. Het kan zijn dat er een andere methode moet worden gevolgd, dus het nalezen van dit bestand is belangrijk.

**NEWS:** bevat vaak een overzicht van de releases uit het verleden met vermelding van uitbreidingen, bugfixes.

**COPYING:** Bij GNU software is het copying bestand aanwezig. Het bevat de General Public License (GPL) en dicteert de licentie waaronder de software aan u ter beschikking is gesteld. Het komt er op neer dat u het recht heeft om de software te bestuderen, te compileren, te her-distribueren en te wijzigen. Er wordt geen garantie gegeven voor de correcte werking. De enige plicht die u heeft is dat, als u deze software verder distribueert deze altijd vergezeld moet gaan van (eventueel gewijzigde) broncode.

**Het compileren.** Eenmaal uitgepakt is het zaak de software te gaan compileren. Als eerste kan men een **.configure** uitvoeren, vervolgens een **make** en daarna een **make install**.

# **./configure** als er een configure bestand staat dan met dit uitgevoerd worden. Om problemen met het PATH te voorkomen dient hier altijd **./configure** als commando te geven. Vervolgens gaat het script kijken of aan alle voorwaarden voor de compilatie wordt voldaan. Bestaan de juist complieerprogramma's wel? Zijn de goede bibliotheken aanwezig? etc. Afhankelijk van het programma dat u wilt compileren kan deze uitvoer meer of minder opties bevatten. Als alle noodzakelijke ingrediënten worden gevonden verschijnt als laatste een tekst zoals hieronder aangegeven:

```
creating menu/Makefile
creating form/Makefile
creating misc/Makefile
creating test/Makefile
creating c++/Makefile
creating Makefile
```

De **makefile** is hier waar het om draait. Dit bestand dient als **invoer** voor het **make** commando waarmee programma's gecompileerd worden. Als er geen configure-bestand is dan is er meestal wel een Makefile aanwezig. Meestal vindt u al deze informatie in het **INSTALL** bestand. U kunt nu beginnen met het **make** commando. Het kan zijn dat er tijdens het compileren een fout optreedt. Dan wordt het compileren beëindigd. een vaak voorkomende fout is dat een bepaalde include-file niet gevonden wordt. Include-files zijn bestanden die de aanroep van bibliotheekroutines beschrijven. Het ontbreken van een include-file leidt tot de conclusie van make dat het niet weet of de aanroep van de routines correct kan worden uitgevoerd. Het vinden van een include-file kan met het **locate** of **find** commando. Vervolgens kunt u handmatig met de editor de Makefile aanpassen en het pad naar de betreffende include-file opgeven.

**make install.** als alles goed is verlopen dan wordt het make-proces zonder foutmeldingen afgesloten. Het laatste dat u dan kunt doen is een **make install** commando geven. Vervolgens wordt de gecompileerde software met alle data-files en bibliotheken op de relevante plaatsen in het systeem geïnstalleerd.

**102.4 Gedeelde bibliotheken.** Het linux-systeem is in staat om gebruik te maken van gedeelde programma-bibliotheken die shared libraries worden genoemd. Op die manier kunnen programma's die gezamenlijk regelmatig bepaalde routines gebruiken kleiner zijn. Door het onderling delen van bibliotheek-code is het geheugenbeslag ook kleiner. Een **shared library** is vaak te herkennen aan de **extensie .so** (van shared object). U vindt er bijvoorbeeld een aantal in de **/lib** directory. Voor het bepalen van de afhankelijkheden van een programma van shared libraries kan het **ldd** commando worden gebruikt. Een programma dat hier uitvoerig gebruik van maakt is bash.

```
linux:/home/rene # ldd /bin/bash
linux-gate.so.1 => (0xffffe000)
libreadline.so.5 => /lib/libreadline.so.5 (0x40032000)
libhistory.so.5 => /lib/libhistory.so.5 (0x40060000)
libncurses.so.5 => /lib/libncurses.so.5 (0x40068000)
libdl.so.2 => /lib/libdl.so.2 (0x400ad000)
libc.so.6 => /lib/tls/libc.so.6 (0x400b1000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Al eerste staat steeds de bibliotheek-naam vermeld gevolgd door de locatie waar

deze bibliotheek op het systeem is geïnstalleerd. Maar hoe weet het systeem dat? Het **ldconfig** loopt alle directories door die in **etc/ld.so.conf** zijn opgegeven. Het maakt indien nodig links aan naar de meest recente versie en bouwt een cache op met de gevonden informatie. Dat gebeurt in dus weer in **/etc/ld.so.conf**

```
linux:/home/rene # cat /etc/ld.so.conf
/usr/X11R6/lib/Xaw95
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/i486-linux/lib
/usr/i486-linux-libc5/lib=libc5
/usr/i486-linux-libc6/lib=libc6
/usr/i486-linuxaout/lib
/usr/i386-suse-linux/lib
/usr/local/lib
/usr/openwin/lib
/opt/kde/lib
/opt/kde2/lib
/opt/kde3/lib
/opt/gnome/lib
/opt/gnome2/lib
include /etc/ld.so.conf.d/*.conf
```

Het kan wel eens gebeuren dat een programma niet kan opstarten en klaagt over de afwezigheid van een of andere bibliotheek. Vooral na het installeren kan dit nog wel eens voorkomen. Dat gebeurt als de betreffende bibliotheek wel is geïnstalleerd maar dat **/etc/ld.so.conf** nog niet is bijgewerkt. **ldconfig** moet dit probleem oplossen. Ook kan het gebeuren dat u een pakket als GNOME heeft geïnstalleerd maar de bibliotheek-directory nog niet in **/etc/ld.so.conf** staat vermeld. Nadat u deze directories heeft toegevoegd aan dit bestand en weer **ldconfig** heeft uitgevoerd, moet alles weer naar behoren werken. Als een programma daarna nog steeds klaagt over de afwezigheid van een bibliotheek is het zaak om de juiste te achterhalen (vanaf de installatie cd-rom of van het internet) en deze te installeren. Daarnaast is de environment variabele **LD\_LIBRARY\_PATH** van belang. De directories die hier worden vermeld, worden bij het zoeken naar gedeelde bibliotheken als eerste doorlopen, daarna volgen de directories **/lib** en **usr/lib** en daarna de bibliotheken in **etc/ld.so.conf**.

## 102.6 RED HAT PAKKETMANAGEMENT

Dit werkt hoofdzakelijk met een commando: **rpm**. Het programma maakt gebruik van een lokale database (**/var/lib/rpm**) waarin alle informatie over de aanwezige pakketten zit opgeslagen. Installatie van een rpm bestand is erg eenvoudig. U gebruikt de optie **-i** van **install**.

```
#rpm -i xyz.rpm installeert het pakket xyz
#rpm -e xyz.rpm (van erase) verwijdert het pakket xyz
#rpm -U xyz.rpm (update) voert een update uit van de pakketten xyz.
```

U kunt ook het **rpm** commando gebruiken om informatie te krijgen over de geïnstalleerde pakketten. Dat doet u door als eerste de optie **-q** (query) te geven. Vervolgens kunt u allerlei opties toevoegen om deze informatie verder te stroomlijnen. Als u wilt weten welke rpm pakketten er allemaal op uw systeem zijn geïnstalleerd, geeft u het volgende commando **rpm -qa**. Er verschijnt dan een enorm lange lijst. In combinatie met het **grep** commando kunt u vrij snel zien of een bepaald programma is geïnstalleerd.

```
linux:/home/rene # rpm -qa | grep koffice
koffice-spreadsheet-1.3.3-3
koffice-1.3.3-3
koffice-il8n-nl-1.3.3-4
koffice-presentation-1.3.3-3
koffice-illustration-1.3.3-3
koffice-wordprocessing-1.3.3-3
```

```
rpm -qi koffice-1.3.3-3 geeft algemene informatie over koffice
rpm -ql koffice-1.3.3-3 geeft de lijst van bestanden die het pakket bevat
rpm -qc koffice-1.3.3-3 geeft de configuratie-scripts van koffice
```

Het is ook mogelijk om te achterhalen welk reeds geïnstalleerd bestand van welk pakket aanwezig is. Dat gebeurt met optie **-f**.

```
linux:~ # rpm -qf /etc/issue
suse-release-9.2-3
```

Natuurlijk zijn bovenstaande programma's leuk, maar hoe komt u er achter in welk nog niet geïnstalleerd pakket een bepaald bestand zit dat u zoekt. Dan kan door

aan de query opties de optie -p toe te voegen gevolgd door de bestandsnaam van het betreffende bestand. Zo kunt u ook met behulp van een wildcard \* en het grep commando zoeken welk rpm bestand een bepaald bestand in zich heeft. Dan krijgt men het commando zoals het volgende: **#rpm -qlp \* | grep gezocht bestand.**

In de directory **/usr/lib/rpm** staan een groot aantal bestanden die door rpm gebruikt worden. Dit zijn o.a scripts die door rpm worden gebruikt en waarin macro's staan gedefinieerd. Ook bevat deze directory een **rpmrc** bestand, dat definities bevat van een groot aantal variabelen die in het rpm proces gebruikt worden. Het wijzigen van dit bestand is zinloos omdat het met elke nieuwe versie wordt overschreven. Specifieke aanpassingen worden opgeslagen in het bestand **/etc/rpmrc** [systeembreed] of **~/.rpmrc** [lokaal]. [op SUSE zijn deze bestanden niet te vinden!]

Boek 1. Hoofdstuk 2 bladzijde 23 t/m 42  
103.1 effectief werken met de UNIX commando regel (5)  
103.2 tekststromen verwerken door gebruikmaken van filters (6)

Boek 1 hoofdstuk 4 bladzijde 65 t/m77  
103.3 Het kunnen uitvoeren van basisbestandsbeheer (3)  
103.4 Het gebruik van streams, pipes en redirection (5)

Boek 1 Hoofdstuk 5 bladzijde 81 t/m 90  
103.5 Procesbeheer (5)  
103.6 Het aanpassen van procesuitvoeringsprioriteiten (2)

Boek 1 Hoofdstuk 3 bladzijde 65 t/m 77  
103.7 Zoekopdrachten en reguliere expressies (3)  
103.8 Eenvoudig tekstverwerken met VI (1)

**103.1 Shell:** de schil (shell) tussen de gebruiker en het systeem. De shell is een programma waarmee u commando's kan invoeren. De meest gebruikte shell in Linux is de **BASH**-shell (BASH: Borne Again Shell). U kan achter de prompt het commando invoeren. `rene@linux:~>` (=prompt) Met het commando **"exit"** verlaat u BASH. **"Logout"** kan ook, maar alleen, als u de sessie ook zo bent begonnen, dan fungeert het als het afsluiten van het gehele systeem.

**Directories.** De harddisk is ingedeeld in meerdere directories, elke directory kan weer sub-directories bevatten. En die laatste kunnen ook weer sub-directories bevatten. Als dit systeem wordt uigetkend ontstaat er een omgekeerde boomstructuur. De hoofd-directory is dan de **root**. U staat echter doorgaans nadat u bent ingelogd in uw eigen home-directory. Het commando **"pwd"** (print working directory) laat dit zien.

Het commando **"cd"** (change directory) + die andere directory moet u geven om naar die andere directory te gaan. Bv. `cd /usr/share/doc`. Zonder argument krijgt u altijd uw home-directory.

```
- home    directory:  cd    (home directory)
- actuele directory: cd .  (actuele niveau)
- parent  directory: cd .. (niveau hoger)
```

Met het commando **"man" + argument** kan er hulp worden gevraagd. **"man pwd"** geeft een handleiding van het commando `pwd`. Het commando **"ls"** (list) geeft aan welke zichtbare bestanden in de working directory zijn. Er zijn ook niet-direct zichtbare bestanden. Door middel van de optie **-a** (all) aan **ls** worden ook deze op het scherm afgebeeld.

Het verschil tussen zichtbare en onzichtbare bestanden is de punt die ervoor staat. Deze **.bestanden** bevatten voornamelijk instellingen die voor de gebruiker van toepassing zijn.

Commando's kunnen aan elkaar worden geplakt door ze achter elkaar te zetten en ze vervolgens te scheiden door middel van het **;** teken. `cd .. ; ls ; pwd` Er worden hier drie commando's gegeven. U schuift een directory hoger, er wordt een lijst gegeven welke bestanden aanwezig zijn en de working directory wordt op het scherm getoond. Mocht een opdrachtregel te lang zijn dan kan die door u worden afgebroken door middel van het teken **\** Dit betekent dan dat alle volgende commando's achter elkaar worden ingelezen, en verwerkt.

```
Dus cd ; \
> pwd ; \
> ls
= hetzelfde als: cd ; pwd ; ls
```

**Omgeving (environment).** De shell werkt in een gedefinieerde omgeving. Door middel van **"set"** of **"env"** commando krijgt u de variabele en haar inhoud. Het is gebruik om de **variabele in hoofdletters** te typen, en de **inhoud** daarvan in **kleine**

**letters.** Een paar voorbeelden:

```
BASH=/bin/bash
KDEDIR=/opt/kde
MAIL=/var/spool/mail/cursus
MANPATH=/usr/local/man:/usr/share/man:/usr/X11R6/man:/opt/gnome/share/man
PWD= home/cursus
USER=cursus
WINDOWMANAGER=/opt/kde3/bin/startkde
XCURSOR_THEME=crystalwhite
XDM_MANAGED=/var/run/xdmctl/xdmctl-:0,maysd,mayfn,sched,rsvd,method=classic,auto
```

Variabelen kunnen ook **zelf** worden **gedefinieerd**. Het definiëren van een environment variabele is erg eenvoudig. Door middel van de variabele NAAM gekoppeld met de inhoud piet. NAAM=piet. U kunt de inhoud van de variabele bekijken door er een \$ voor te zetten. Een voorbeeld is het echo commando: echo \$NAAM geeft piet. **echo \$PATH** geeft **/usr/local/bin:/usr/enz./**

**Variabelen exporteren.** Environment-variabelen kunnen worden geëxporteerd. Dit gebeurt met het export-commando: **#export NAAM**. Hierdoor wordt de variabele "NAAM" met zijn inhoud beschikbaar gesteld aan programma's die vanuit deze shell worden gestart. Een paar belangrijke:

```
PATH: deze variabele bevat alle directories die worden doorzocht als u een commando heeft ingetypt.
PATH=/home/rene/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gnome/bin:/opt/kde3/
bin:/usr/lib/java/jre/bin
HOME: de naam van de home directory van deze gebruiker.
HOME=/home/rene
PS1: bevat de variabele van de standaardprompt. Deze variabele bevat zelf ook weer een aantal variabelen. Zo wordt
de tekst \u vervangen door de actuele gebruiker, \h door de systeem-naam en \w door de actuele directory.
PS1='\u@\h:\w> '
PRINTER: naam van de standaardprinter voor deze gebruiker. (nog niet gedefinieerd)
```

Als men een programma wil starten welke niet in een directory staat die door PATH is gedefinieerd kan men zelf het PATH opgeven. Stel dat men een prg in zijn home directory heeft staan (en deze directory is niet in het PATH vermeld) dan typt men /home/cursus/prg. Dit kan weer worden afgekort door **./prg**

**unset** Dit commando kan gebruikt worden om variabelen en/of eigenschappen te verwijderen.

**De commando-historie:** de bash-shell kent een commando-historie. Deze bevat standaard de 500 laatste commando's die u heeft gegeven. Met de toets pijltje omhoog geeft het voorlaatste commando. Met het pijltje omlaag loopt u weer terug naar het voorlaatste commando. Het commando **history** geeft alle commando's weer. Het commando kan weer gestart worden door een uitroepteken ! met het commando-nummer. Bv !432 Met de **tab** kan de opdrachtregel worden aangevuld. Als BASH het niet direct weet dan zijn meerdere opties mogelijk. Door dan nog een keer op tab te drukken krijgt u alle mogelijkheden.

U kan **zelf commando's** definiëren. Als u vaak een lang commando invoert is het handig om daar een kortere variant voor te bedenken. Dat noemt men **aliases**. De actuele lijst van alle aliases krijg je via het commando "alias". Bv. alias ll='ls -l' of **alias rd='rmdir'**. Structuur: alias <nieuwe gemaakte korte commando>='<zelfde lange commando>'

**whoami:** oftewel onder welke gebruikersnaam ik op dit moment bent ingelogd.  
**type:** dit commando is erg makkelijk om te kijken waar een bepaald programma zich bevindt als het ergens in het zoekpad staat. rene@linux:~> type kde  
kde is /usr/X11R6/bin/kde

**exec:** exec start het hierna genoemde programma op bv **exec knode**. [Suse niet]  
**file:** probeert een inschatting te maken om wat voor bestand het gaat. rene@linux:~>

```
file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.5, dynamically linked (uses
shared libs), not stripped. Dus een uit te voeren (execute) programma-bestand.
```

**~/bash\_history** Hierin staan de bash commando's die de lokale gebruiker heeft gegeven. Dit bestand is met behulp van cat uit te lezen.

Het aanpassen van de omgeving van individuele gebruikers kan op een linuxsysteem op meerdere plaatsen. Ten eerste in het bestand **/etc/profile**. Dit bestand wordt als eerste uitgevoerd zodra iemand zich aanmeldt op het systeem. Vervolgens wordt in de eigen home-directory gekeken of daar een individueel bestand staat: **.profile** In dit laatste bestand kunt u de omgeving voor de betreffende gebruiker aanpassen. Afhankelijk van de lees- en schrijfrechten die het bestand heeft is de gebruiker zelf in staat om dit bestand aan te passen.

In het **/etc/profile** bestand wordt een groot aantal zaken voor de gebruiker geregeld. Elke distributie kan anders zijn. hieronder vindt u een voorbeeld van een gedeelte van een profilebestand. Het commentaar wordt altijd aangegeven door een #. U kunt zelf met vi dit bestand bewerken.

```
rene@linux:~> cat /etc/profile
# /etc/profile for SuSE Linux
#
# PLEASE DO NOT CHANGE /etc/profile. There are chances that your changes
# will be lost during system upgrades. Instead use /etc/profile.local for
# your local settings, favourite global aliases, VISUAL and EDITOR
# variables, etc ...

# Check which shell is reading this file
#
if test -f /proc/mounts ; then
  case "`/bin/ls -l /proc/$$/exe`" in
    */bash)    is=bash ;;
    */rbash)   is=bash ;;
    */ash)     is=ash  ;;
    */ksh)     is=ksh  ;;
    */zsh)     is=zsh  ;;
    */*)       is=sh   ;;
  esac
else
  is=sh
fi

#
# Initialize terminal
#
tty=`tty 2> /dev/null`
test $? -ne 0 && tty=""
if test -O "$tty" -a -n "$PS1"; then
  test -z "${TERM}"      && { TERM=linux; export TERM; }
  test "${TERM}" = "unknown" && { TERM=linux; export TERM; }
  # Do not change settings on local line if connected to remote
  if test -z "$SSH_TTY" ; then
    test -x /bin/stty    && /bin/stty sane cr0 pass8 dec
    test -x /usr/bin/tset && /usr/bin/tset -I -Q
  fi
  # on iSeries virtual console, detect screen size and terminal
  if test -d /proc/iSeries -a \( "$tty" = "/dev/ttyl" -o "$tty" = "/dev/console" \) ; then
    LINES=24; COLUMNS=80; export LINES COLUMNS TERM
    test -x /bin/initviciocons && { eval `/bin/initviciocons -e`; }
  fi
fi
unset TERMCAP

#
# Time until a complete key sequence must have arrived
#
#ESCDELAY=2000
#export ESCDELAY

#
# The user file-creation mask
#
umask 022

# ksh/ash soemtimes do not know
#
test -z "$UID"  && UID=`id -ur 2> /dev/null`
test -z "$EUID" && EUID=`id -u 2> /dev/null`
test -z "$USER" && USER=`id -un 2> /dev/null`
test -z "$MAIL" && MAIL=/var/spool/mail/$USER
test -z "$HOST" && HOST=`hostname -s 2> /dev/null`
test -z "$CPU"  && CPU=`uname -m 2> /dev/null`
test -z "$HOSTNAME" && HOSTNAME=`hostname 2> /dev/null`
# ... en er wordt nog veel meer aangegeven.
```

**Commando-opbouw.** Een linux commando bestaat altijd uit drie onderdelen: [commando-naam] [opties] [argumenten]. U kunt een commando zonder opties en argumenten aangeven. Opties zijn herkenbaar aan het feit dat ze met een - of

twee -- streepjes beginnen. Met twee streepjes wordt een lange naam verwacht. Dus **commando -h = commando --help**

### 103.2 tekststromen verwerken door gebruikmaking van filters

**cat**: zal de inhoud van het bestand in het scherm afbeelden. optie **-n** zet een regelnummer voor de uitvoer.

**tac**: is het omgekeerde van het commando cat. De inhoud van het bestand zal worden afgebeeld, maar nu beginnend met de laatste regel

**nl**: voorziet het bestand van regelnummers

**more**: om de inhoud van een bestand met een scherm tegelijk te raadplegen. Als de uitvoer groter is dan een scherm dan wordt aangegeven hoeveel procent van de tekst getoond is. Een druk op de spatiebalk geeft dan het volgende scherm weer.

**less**: met dit commando kunt u met pijltjestoetsen en PgUp en PgDn makkelijk en snel door een bestand heen gaan. (less is more! -> handiger)

**tail**: laat de laatste regels van een bestand zien. Met de optie **-5** bv worden de laatste vijf regels gegeven. **tail -5 bestand**

**head**: omgekeerde van tail. **head -5 bestand** geeft derhalve de eerste vijf regels van bestand.

**wc**: word count. Geeft het aantal regels, woorden en aantal letters.

**grep**: grep is in staat om een bestand te doorzoeken op de aanwezigheid van een opgegeven tekst

**sort**: om in bestanden te sorteren. In zijn meest simpele vorm sorteert het een bestand uitgaande van de meeste links tekst. Het is ook mogelijk, handig bij namen, te sorteren op het tweede woord in de regel. Er wordt met tellen met 0 begonnen. Dus **sort +1 bestand** sorteert op het tweede woord in de regel.

**cut**: het snijdt een gedeelte uit een bestand. Met de optie **-f** kunt het gewenste veld opgeven.

**od**: dit commando od geeft de octale inhoud van het bestand. **od -h** geeft het bestand hexadecimaal aan.

**split**: dit commando kan worden gebruikt om het bestand te splitsen in een aantal resultaat-bestanden van ieder een bepaalde maximaal.

**fmt bestand**: om de tekst opnieuw te formateren. Alle tekst wordt achter elkaar gezet. Met de **w** optie (**-w**) kunt u de maximale tekstbreedte opgeven.

**expand**: een bestand inlezen en de tekst uitvoeren

**unexpand**: doet precies het omgekeerde van expand en zet dus een reeds spaties in een bestand weer om in 1 of meer tabs

**uniq**: verwijdert regels uit een bestand die meerdere keren achter elkaar voorkomen. U kunt daarbij aangeven hoeveel velden en/of karakters genegeerd moeten worden voordat regels worden vergeleken, en of u het aantal gevonden identieke regels aan het begin van elke uitgevoerde regel vermeld wilt zien.

**paste**: dit commando dient voor het samenvoegen van twee bestanden. Als u met paste twee bestanden samenvoegt wordt de eerste regel van de uitvoer gevormd door de eerste regel van het eerste bestand, en na wat spaties, de eerste regel van het tweede bestand.

**join bestand**: is wat gecompliceerder dan het paste-commando. Het verbindt namelijk twee bestanden met elkaar op basis van het eerste veld. Die regels waarvan het eerste veld met elkaar overeenkomen wordt samen afgebeeld.

**pr bestand**: dient om uitvoer op te maken voor een regelprinter. Het bestand wordt uitgelijnd, krijgt een koptekst en een formaat van de pagina.

**tr** bestand: dient voor het vervangen (transform) of verwijderen (**-d**) van tekens uit een bestand.

Als u bv. de 'a' uit het bestand wilt verwijderen: **tr -d 'a' < bestand**

Het **<** teken staat ervoor dat de output van bestand de input van het commando **tr** is, dit heet redirection.

**sed**: staat voor streameditor, het is een commando waarmee de stroom gegevens wordt bewerkt terwijl deze langskomt. Het is als het ware een uitgebreide versie van "tr". **sed -e 's/a/x' bestand**. Wordt door door sed optie **-e** (editor) 's' (substitueert) a in een x'

### 103.3 Het kunnen uitvoeren van basisbestandsbeheer

Het **ls** commando wordt gebruikt om informatie van bestanden af te beelden. "ls

-al" beeldt de onzichtbare bestanden ook af.

```
ls -al      1      2 3      4      5      6      7
-rw-r--r-- 1 cursus users 795 jul 7 11.35 bestand
De 1st kolom geeft de status van de toegangsrechten van het bestand
De 2de kolom geeft het aantal referenties (links) naar het bestand weer
De 3de kolom geeft de eigenaar van het bestand aan.
De 4de kolom geeft gebruikers waartoe het bestand toe behoort
De 5de kolom geeft het aantal bytes (grootte) van het bestand aan
De 6de kolom geeft de datum en tijd aan
De 7de kolom geeft de naam van het bestand aan (hier dus bestand)
```

**cp.** Wordt gebruikt om bestanden mee te kopiëren. In de meest eenvoudige vorm. "cp bestand bestand2". Er kunnen meerdere bestanden worden gegeven, als de laatste een directory wordt alles in die directory gekopieerd. Zo zal cp standaard zonder enige melding een bestaand bestand overschrijven. Met optie **-i** wordt u wel gewaarschuwd. Ook is het mogelijk de gehele directory mee te kopiëren. De optie daarvoor is **-a** Daarbij wordt geprobeerd de originele bestandsinformatie in stand te houden.

**mv.** Wordt gebruikt voor de move-operatie. Dit kan het verplaatsen van een bestand of een directory naar een directory zijn. Maar het kan ook het verplaatsen van een bestand naar een andere naam in dezelfde directory zijn. Dan komt het neer op het herbenoemen van een bestand.

**mkdir.** Wordt gebruikt om een nieuwe directory aan te maken. De werking is vrij simpel. Gewoon "mkdir directorynaam" intypen.

**rm.** Wordt gebruikt om bestanden te verwijderen (remove) Met de optie **-r** kunt u ook een directory met inhoud wissen. Het is dan echt weg!

**rmdir.** Verwijdert de directory. Alleen als de map leeg is, als er geen bestanden in staan, wordt deze verwijderd. commando "**rm -r** directorynaam" werkt dan **sneller**.

**find** kijkt wat af van de gebruikelijke manier van invoeren. De directory waarin gezocht moet worden. De naam van het gezochte bestand. En wat er mee moet gebeuren. Zo kan men bestanden vinden die voor een bepaalde datum zijn gemaakt, of niet groter dan een bepaald formaat zijn. Komt later terug  
**touch.** U kunt een nieuw (leeg) bestand aanmaken met **touch bestand** Dit bestand heeft verder geen grootte, is leeg.

**file globbing:** Maskers/wildcards De term masker is een veel gebruikte vertaling van het Engelse begrip "pattern". Met patterns kunt u op een makkelijke manier een selectie van van bv bestanden definiëren. De bash shell kent de volgende mogelijkheden.

- \* komt overeen met elke tekst (ook lege tekst)
- ? komt overeen met een enkel willekeurig karakter
- [...] komt overeen met elk van de ingesloten karakters bv [a-z,A-Z]

**103.4 Streams, pipes en redirection.** Linux kent de mogelijkheid om de uitvoer van programma's van richting te laten veranderen. Dit noemt men **redirecten**.  
cat bestand > uitvoer.

Het cat bestand wordt met het > teken gebruikt als invoer voor het bestand uitvoer. Als u de invoer wil bijvoegen aan een al bestaand document dient u twee keer het > teken te geven. **ls -al >> uitvoer** voegt de nieuwe informatie toe aan het bestand uitvoer (de cat informatie en de ls informatie staat nu in het uitvoer bestand)

**Invoer-redirection.** Bij **sed -e 'y/in/rb/' < bestand**. De inhoud van bestand wordt gebruikt als **input** om een sed bewerking te ondergaan. Alles van "in" wordt in het bestand veranderd in "rb".

<< Dit type redirection instrueert de shell om invoer te lezen totdat een opgegeven woord wordt gevonden. Dit wordt ook wel here-document genoemd.

**Pipes.** Met behulp van het pipe-teken | kunnen commando's achter elkaar worden

```
gezet. cat ls -al | sed -e 'y/-/+/' | sort +6
```

De uitvoer van `ls` wordt gebruikt door `sed` die `-` in `+` veranderd en deze uitvoer dient weer als invoer voor `sort +6`, die sorteert op de zesde kolom.

**In- en uitvoerkanalen.** In bovenstaand voorbeeld wordt dus iets veranderd in de in- en uitvoer van commando's. Linux kent drie kanalen:

**stdin:** standaard invoer (input), kanaal 0

**stdout:** stand uitvoer (output), kanaal 1

**stderr:** standaard fout (error), kanaal 2

U kan deze **kanaalnummers** ook gebruiken om de uitvoer te **redirecten**.

Zo zal `ls 1>t1` uitvoer `ls`, kanaal 1 naar `t1` worden geschreven (denk erom hier dienen geen spaties te worden gegeven) Hiermee kunt u ook in andere gevallen foutmeldingen laten wegschrijven naar een bestand. `rm dir1 2>t1` De foutmelding die `rm dir1` geeft, namelijk de directory is niet leeg oid, wordt naar bestand `t1` geschreven.

Het is ook mogelijk om de uitvoer en de foutmeldingen naar hetzelfde bestand te sturen. U kunt natuurlijk het commando `rm dir1 1>t1 2>t1` geven maar er is een kortere oplossing. Het teken `&` kan gebruikt worden als variabele. Het commando `rm dir1 1>t1 2>&l` betekent dat de foutmeldingen in het eerstgenoemde bestand komen.

**xargs.** Hiermee kunt u de uitvoer gebruiken als invoer.

```
cat < invoer geeft het bestand
```

```
cat invoer | xargs cat geeft een cat van het bestand.
```

**tee.** Het is soms handig als u data-stromen kunt splitsen. Het ene deel wordt opgeslagen in een bestand, terwijl het toch verder bewerkt wordt. Dat kan met het "tee" commando.

```
cat bestand | sed -e 'y/a/b/' | tee file1 | sed -e 'y/o/e/' > file2
```

In het `cat bestand` worden de letters `a` veranderd in `b` en deze nieuwe informatie wordt weggeschreven in bestand `file1`. De informatie `file1` wordt verder bewerkt waarbij alle letters `o` veranderen in de letter `e`. De output hiervan wordt weggeschreven in `file2`.

### 103.5 Het beheren van processen

In de basis bestaat linux slechts uit een paar onderdelen. een daarvan is de kernel, de kern van het systeem, en de andere twee zijn bestanden en processen. Elk programma is een proces en bewerkt op zijn beurt weer een of meer bestanden. De kernel is bij dit alles de grote scheidsrechter die bepaalt wanneer welk proces wat mag doen. Een van de dingen die u regelmatig zult willen doen is het afbreken van de uitvoering van een programma. Een druk op `Ctrl-C` is meestal voldoende. Een aantal programma's zijn zo gebouwd dat zij niet reageren op deze toets-combinatie. U moet die dan anders zien af te breken.

Processen kunnen in de voorgrond draaien. Zo kunt een een programma starten, dan wordt er naast de shell een tweede proces gestart. Een is daarvan op de voorgrond actief aanwezig en gekoppeld aan een terminaal. Een ander proces is dan actief op de achtergrond en niet gekoppeld aan een terminaal. Het starten van een proces in de **achtergrond** kan simpelweg door het commando te eindigen met `&`.

Elk proces heeft een uniek nummer, het proces-id (**PID**). Met dit nummer regelt de kernel alle bewerkingen rondom het proces. Met dit PID kan men bv processen stoppen, vertragen, versnellen en herstarten.

Met het commando "**jobs**" kunt u een overzicht krijgen vanuit deze shell in de gestarte processen (die in de achtergrond staan). De nummers die jobs geeft zijn niet PID's van processen maar eigen volgnummers. Het is mogelijk om een draaiend proces van de voor- naar de achtergrond te verplaatsen.

U kunt een werkend proces onderbreken door **Ctrl-Z** te drukken. Op dat moment wordt het werkende proces in de achtergrond geplaatst (het draait dan niet verder) en u krijgt de shell-prompt in beeld.

Voor het weer naar de voorgrond brengen van een programma kunt u het **fg** (foreground) + jobs-nummer gebruiken. Met **bg** commando kunt u daarna ook weer een gestopt proces in de achtergrond laten draaien (dat heeft hetzelfde effect als het opstarten met &). Merk op dat hier de nummering wordt gebruikt zoals het jobs-commando die weergeeft en niet de proces ID's. Als u een proces in de achtergrond heeft gestart en vervolgens uitloopt terwijl dit proces nog actief is, dan wordt dit proces door een actie van de shell beëindigd. Dit kan wel eens een ongewenst effect zijn. Daarvoor is het **nohup** commando (van no hang up) in het leven geroepen. Het proces wordt dan gestart en loopt nu door, ook als u uitloopt uit het systeem.

**Systeemprocessen.** Als beheerder van linux moet u soms ook andere processen, dan die vanuit de shell, beheren. Belangrijk is natuurlijk om een overzicht te krijgen van de binnen het systeem actieve processen. Het **ps** commando dient voor het verkrijgen van de draaiende processen in het algemeen. rene@linux:~> ps

```
PID TTY          TIME CMD
5926 pts/2      00:00:00 bash
5935 pts/2      00:00:00 ps
```

Het programma geeft een lijst door van de door u gestarte processen, te weten de shell "bash" en het "ps" commando. Dit is natuurlijk maar een klein overzicht van de actieve processen op het gehele systeem (man ps). opties -aux geeft meer.

**TOP.** Het commando "top" is een programma dat een continue lijst geeft van draaiende processen. Deze kunnen worden gesorteerd op gebruikte CPU tijd, processor-tijd en PID. Standaard ververs het programma zijn gegevens elke 5 seconden, maar dit kan veranderd worden. Met "h" krijgt u de mogelijke opties. Er wordt aangegeven hoeveel procent tijd de processor waarmee bezig is (user en system). Het idle-percentages geeft aan hoeveel procent van de tijd de processor niets staat te doen. Verder het geheugengebruik: de hoeveelheid intern geheugen; daarna hoeveel daarvan gebruikt wordt door applicaties. Hoeveel er nog van over is (free), hoeveel geheugen door de programma's wordt gedeeld (shared). En hoeveel geheugen wordt gebruikt als buffer. De onderste van de drie regels geeft informatie over de hoeveelheid swap-ruimte. Een overzicht van processen is handig, maar er moet ook een methode zijn om de werking van processen te stoppen. Dat gebeurt met het "kill" + PID commando. Door de kernel wordt met behulp van het kill-sigitaal (SIGKILL =9) het betreffende proces verwijderd.

rene@linux:~> top

```
top - 16:10:18 up 1:29, 3 users, load average: 0.08, 0.11, 0.09
Tasks: 68 total, 1 running, 67 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.0% us, 0.7% sy, 0.0% ni, 98.0% id, 0.0% wa, 0.3% hi, 0.0% si
Mem: 190904k total, 184768k used, 6136k free, 8360k buffers
Swap: 380984k total, 748k used, 380236k free, 77300k cached
```

```

PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
6070 rene     16   0 30560 16m  27m  S  1.0  9.0   0:01.26 kdeinit
3946 root      15   0 79324 10m  68m  S  0.7  5.8   0:41.37 X
   1 root      16   0   588  160  444  S  0.0  0.1   0:05.44 init
   2 root      34  19     0   0    0  S  0.0  0.0   0:00.00 ksoftirqd/0
   3 root      5 -10     0   0    0  S  0.0  0.0   0:00.07 events/0
   4 root      5 -10     0   0    0  S  0.0  0.0   0:04.43 kacpid
   5 root      5 -10     0   0    0  S  0.0  0.0   0:00.04 kblockd/0
   6 root      5 -10     0   0    0  S  0.0  0.0   0:00.00 khelper
   7 root     15   0     0   0    0  S  0.0  0.0   0:00.00 pdflush
   8 root     15   0     0   0    0  S  0.0  0.0   0:00.18 pdflush
  10 root      8 -10     0   0    0  S  0.0  0.0   0:00.00 aio/0
   9 root     16   0     0   0    0  S  0.0  0.0   0:00.22 kswapd0
 162 root     16   0     0   0    0  S  0.0  0.0   0:00.00 kseriod
 293 root      5 -10     0   0    0  S  0.0  0.0   0:00.00 reiserfs/0
 466 root      6 -10     0   0    0  S  0.0  0.0   0:00.00 kcopyd
```

Enzovoorts. Met top -u rene wordt er een overzicht gegeven van alle processen van gebruiker (user) rene

**103.6 Het versnellen en vertragen van processen.** Alle processen hebben een bepaalde prioriteit. Deze krijgen ze tijdens het opstarten toegewezen, maar de prioriteit kan in een later stadium worden gewijzigd. De prioriteits-waarde is te zien als een bepaalde verdragingsfactor. Normaal gesproken krijgt een proces

de waarde 0 toegekend. De waarde kan variëren van -20 tot 19. Een normale gebruiker mag de waarde van een proces alleen verhogen (wordt dus trager, en is nice voor het systeem!). De waarde van een proces verlagen, het proces wordt dan **sneller**, mag alleen de **root**. **ps -yl** geeft o.a de NI (waarde). Met het commando **renice + nieuwe prioriteits-waarde \* PID** kunt u het **proces vertragen/versnellen**.

```
rene@linux:~> ps -yl
S  UID  PID  PPID  C  PRI  NI  RSS   SZ  WCHAN  TTY          TIME CMD
S 1000 6083 6070  0  76   0 1740 1001 wait4 pts/1      00:00:00 bash
R 1000 6290 6083  0  77   0  648  541 -      pts/1      00:00:00 ps
rene@linux:~> renice 6083 -19 renice: -19: slechte waarde (mag als niet-root-
zijnde niet het proces versnellen - bash/6083)
```

### 103.7 Het verrichten van zoekopdrachten door tekstbestanden en daarmee gebruik maken van reguliere expressies [sed, grep en regexp gebruik]

Een reguliere expressie is een patroon waarmee een aantal woorden wordt beschreven, zoals bv het patroon 'a' alle woorden beschrijft die met een 'a' beginnen. Een reguliere expressie kan worden gebruikt in combinatie met bv grep of sed. De laatste twee verrichten daadwerkelijk de wijziging van de ingevoerde tekst, terwijl de reguliere expressie aangeeft welke woorden of teksten er gewijzigd worden. Een lijst van karakters tussen [en] betekent dat elk karakter uit de lijst mag voorkomen, als deze lijst begint met een ^ betekent dat geen enkel karakter uit de lijst mag voorkomen.

De . staat (punt) staat voor elke willekeurige karakter. De ^ betekent het begin, en de \$ betekent het einde van de regel. Een reguliere expressie mag gevolgd worden door een of meer herhalings-indicatoren. Het teken ? betekent dat de voorgaande reguliere expressie optioneel is en ten hoogste een keer mag matchen. Het teken \* betekent dat de voorgaande reguliere expressie nul of meer keer mag worden gematched. Het teken [n] betekent dat de voorgaande tekst precies n keer gevonden moet worden.

[123] betekent dat alle woorden die beginnen 1, 2 of 3  
[^ab] betekent dat alle woorden die niet met a of b beginnen  
^[aA] betekent alle regels die met a of A beginnen

Het commando `grep ^[aA] *` betekent: Laat van alle bestanden die regels zien die met een a of A beginnen.

**Regexp** Op sommige systemen komt dit commando voor. Dit zal een string vergelijken op basis van een opgegeven reguliere expressie en vervolgens een 1 als retourwaarde geven als een deel overeenkomt en een 0 als het niet het geval is. Ook find kent een optie regexp waarmee men zoekopdrachten kan definiëren op basis van een reguliere expressie.

**103.8 De vi editor.** De vi editor kent twee verschillende modes. De invoermode en de commando-mode. De invoer-mode heeft de I (insert), A (append) en de O (?) toets. Met "**vimtutor**" krijgt u een mini-cursus. Door middel van de escape-toets keert u terug naar de commando-mode. In het algemeen dient men een dubbele punt te geven om een commando te geven.

```
:q = stop met vi
:q! = stop met vi en verlaat programma zonder bestand op te slaan
:w = schrijf wijzigingen naar bestand
:wq= sla het bestand op en verlaat vi.
:sav + naam bestand = slaat de tekst op onder de opgegeven naam
ZZ zelfde als :wq maar dan sneller
U kunt door de tekst bewegen door: H J K L [zo staan ze op het toetsenbord]
H regel naar links
J regel naar beneden
K regel naar boven
L regel naar rechts
```

In command-mode:

i = insert: voeg tekst in voor cursorpositie -tekstmode

a = append: voeg tekst in na cursorpositie

dd = delete: verwijdert hele regel

D = verwijdert alle tekst vanaf de cursorpositie tot het einde van het scherm

yy = slaat regel op in buffer

p = paste plakt buffer op de aangegeven regel

o = start met invoer onder de huidige regel

x = verwijder teken op cursorpositie

/woord = start met het voorwaarts zoek van het woord

?woord = start met het achterwaarts zoeken van het woord

#### **104: Devices, filesystemen en FHS (24)**

Boek 1 Hoofdstuk 6 bladzijde 93-112

104.1 Het creëren van partities en filesystemen (3)

- 104.2 Het instandhouden van de integriteit van filesystemen (3)
- 104.3 Het beheren van filesystemkoppelingen (3)

Boek 1 hoofdstuk 7 bladzijde 117-140

- 104.4 Het instellen van bekijken van disk-quota (3)
- 104.5 Toegangsrechten regelen bestanden (5)
- 104.6 Beheren bestandseigendom (1)
- 104.7 Het maken en wijzigen van harde en symbolische links (1)
- 104.8 Vinden bestanden en plaatsen in FHS-directory (5)

Devices vormen een belangrijk onderdeel van het linux besturingssysteem. Het is namelijk de manier waarop een linuxsysteem communiceert met de aangesloten hardware. De kernel doet erg veel op het gebied van communicatie met de aangesloten rand-apparaten. Er zijn drivers van hardware aanwezig in de kernel. De interactie tussen de hardware en gebruikers vindt plaats via device-bestanden. Deze bestanden staan in de **/dev-directory**. Met de opdracht "**ls -la /dev**" verschijnt er een overzicht van een aantal belangrijke bestanden.

Interessante device-files:

- TtySO - de eerste seriële poort van het systeem
- TtyS1 - de tweede seriële poort van het systeem
- fd0 - de eerste diskdrive van het systeem
- hda - de eerste IDE harddisk
- hda1 - de eerste partitie op de eerste harddisk van het systeem
- hdb - de tweede IDE harddisk
- sda - de eerste SCSI harddisk
- tty1 - de eerste terminaal
- zero - een device waar altijd de waarde 0 uitkomt
- random - een device waar een willekeurig getal uit komt
- null - een device waarin u van alles kunt sturen wat vervolgens daarin verdwijnt

In feite bestaan er drie soorten devices en het verschil is te zien aan het eerste karakter met een `ls -l /dev` commando:

**brw-rw-rw-** dit is een **blockdevice**. Een blokdevice is voor gebufferde communicatie. Een programma dat gebruikt maakt van een **harddisk** of diskettestation wil graag gebruik maken van **gebufferde communicatie** want dat komt de efficiëntie ten goede.

**crw-rw----** dit is een **character-device**. Een character-device is voor **ongebufferde communicatie**. Als iemand achter een terminaal werkt die aangesloten is op de tweede seriële poort, dan wil de gebruiker waarschijnlijk niet dat de computer wacht tot dat de buffer vol is voordat de inhoud wordt verwerkt. hij wil dat zodra hij een **toets indrukt, deze ook door het systeem wordt ontvangen**.

Een zeer apart device-bestand is de **FIFO**. Het is een bestand dat door twee programma's gebruikt kan worden om met elkaar te communiceren. Het eerste programma opent een het bestand om naar te schrijven, een ander programma opent het bestand om van te lezen. De tekst FIFO betekent First In First Out. Een FIFO wordt met een **p** gedefinieerd.

De in de /dev aanwezige bestanden wijken bij een `ls -l` commando enigszins af van wat u normaal gesproken ziet.

```
rene@linux:~> ls -l /dev/hda
brw-rw---- 1 root disk 3, 0 2004-04-06 15:27 /dev/hda
```

Het eerste getal na disk wordt **major-device** genoemd en met dit getal (hier 3) wordt aangegeven welke driver in de kernel bij dit device-bestand hoort. Het tweede getal (hier 0) is het **minor-device** nummer. Dit getal geeft binnen de driver aan om welke variant het gaat.

Met het "**mknod**" kunt u zelf device-bestanden aanmaken. De aanroep van dit commando bestaat uit een aantal opties. De opbouw is als volgt:  
mknod naam type major minor. Bv. **mknod /dev/fd0 B 3 0** is dus het commando om de (block)device-file /dev/fd0 te maken.

Een **harddisk** is een unit in de computer waarin een aantal schijven zit met daarop een magnetische laag waarop gegevens worden opgeslagen. Een schijf wordt opgedeeld in een aantal onderdelen. Ten eerste het aantal **tracks** of cilinders. Dit is te vergelijken met het aantal jaarringen van een boom. Elke harddisk heeft op elke schijf een aantal tracks naast elkaar liggen. Elke cilinder of track is weer opgedeeld in **sectoren**. Omdat een harddisk vaak meerdere magnetische schijven kan bevatten wordt ook vaak het aantal lees/schrijfkoppen vermeld, het aantal **heads**. Een harddisk wordt aan-gesloten op een controller. Dit is een stuk elektronica dat de communicatie met de harddisk regelt. We spreken over disks en daaronder wordt een harddisk, diskette, cd-rom speler, tapestreamer, etc onder verstaan. Er zijn twee typen controllers in omloop: **IDE**: is de afkorting van Integrated Drive Electronics. Op de IDE controller kunt u normaal gesproken twee apparaten aansluiten. De eerste heet master, de tweede slave. Moderne PC's hebben meerdere controllers en kunnen dus meer dan twee apparaten aansluiten. **SCSI**: Small Computer Systems Interface. Op een SCSI controller kunt u vaak zeven tot 15 apparaten aansluiten.

Een harddisk is ingedeeld in **partities**. U kunt een harddisk voorzien van een of meer partities. Van oudsher kan een harddisk maar uit vier partities bestaan. Er is later een systeem toegevoegd waardoor een van de vier partities weer kan worden onderverdeeld. Een partitie bestaat uit een drietal onderdelen.

**Primaire partitie**: is de oudste type partitie. Er kunnen maximaal vier partities op een harddisk zijn gedefinieerd. Deze gebruiken dus ook de nummers 1 tot en met 4.

**Extended partitie**: is het nieuwe partitie. u kunt een eerder aangemaakte primaire partitie aanwijzen als zijnde een extended partitie. Deze kunt u vervolgens weer onderverdelen in maximaal twaalf logische partities. Er kan maar een extended partitie op de harddisk zijn gedefinieerd.

**Logische partitie**: is gedefinieerd deel van de extended partitie. Deze gebruiken de nummers vijf en hoger.

Met **fdisk** kunt u de harddisk partitioneren. Uiteraard alleen als u root bent.  
**fdisk /dev/hda2**. Wees hier voorzichtig mee!

**Filesystemen**. De harddisk is ingedeeld in afzonderlijke partities en moet worden voorbereid voor de opslag van gegevens. Dit doet men door middel van een filesystem op een partitie. Een filesystem is te vergelijken met een boekhoudsysteem. Binnen het systeem worden bestanden opgeslagen met hun namen, datum en tijd van ontstaan, toegangsrechten etc. Er zijn verschillende filesystemen die door linux worden ondersteund. Elk heeft zijn voor- en nadelen: msdos, vfat, iso9660, ext2, ext3, ReiserFS bv.

Een filesystem kunt u aanmaken met het **mkfs**. Als u reiser als fs wil dient u mkreiserfs aan te geven. U moet weten welke partitie u van een filesystem wilt voorzien, welke device daarbij hoort, welk file-systeem u wilt hebben. U dient hier root-permissies voor te hebben. Zoals u een filesystem kunt aan maken in een harddisk-partitie, of op diskette.

```
linux:/home/rene # mkfs /dev/fd0
mke2fs 1.34 (25-Jul-2003)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
184 inodes, 1440 blocks
72 blocks (5.00%) reserved for the super user
First data block=1
1 block group
```

8192 blocks per group, 8192 fragments per group  
184 inodes per group

Writing inode tables: done  
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 28 mounts or  
180 days, whichever comes first. Use tune2fs -c or -i to override.

**104.2 Filestysteem controle.** Stel voor dat een systeem door stroomuitval plotseling down is gegaan. Er dient dan een controle plaats te vinden om de inhoudsopgave van het file-systeem weer overeen te laten komen met de inhoud van de harddisk. Bij dit proces kan dus ontdekt worden dat er gegevens verloren zijn gegaan. Dit controle-proces noemt men een filesystem-check en het hiervoor gebruikte commando is **fsck**. Hierbij wordt de inhoud van de gehele disk gecontroleerd. In feite wordt bij iedere start een fscheck gedaan. De inhoud van **etc/fstab** wordt daarvoor ingelezen [meta-data check en gaat dus vrij snel]

**reiserfsck** wordt gebruikt als de partitie reiserfs als filesystem heeft. Maar dan moet u (SUSE) wel root zijn plus de desbetreffende partitie umounten. En dit moet allemaal in single-user mode gebeuren (telinit 1)

**capaciteit controleren.** Als beheerder is het een van uw taken om de beschikbare harddisk capaciteit te controleren. Deze controle kan op een aantal manieren plaatsvinden. Het **df** commando is het meest gebruikte gereedschap voor controle van de beschikbare diskruimte. De uitvoer kan er als volgt uitzien:

```
rene@linux:~> df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/hda2        38687836    3425416   35262420   9% /
tmpfs            95452         8      95444    1% /dev/shm
```

**df -i** geeft ook het aantal gebruikte inodes weer.

Het is prettig als u weet hoeveel ruimte er in de /home directory is en dat er ook nog genoeg inodes beschikbaar zijn. Met het **du** commando krijgt u een overzicht van de gebruikte diskruimte onder een bepaalde directory. De optie **-sh** zorgt ervoor dat u alleen de totalen krijgt en in een leesbare vorm. U mag dan alleen uw eigen home-directory opvragen, tenzij u root-permissies heeft.

Het FAT systeem werkt niet snel maar is wel redelijk veilig. Elke keer als een bestand wordt opgeslagen wordt ook het FAT bijgewerkt. Als de stroom uitvalt kunt u zo weer opnieuw beginnen immers alle data is er nog. Linux werkt echter wat anders. Daar moet een systeem gecontroleerd down worden gebracht omdat alle omdat veel veranderingen uit het geheugen nog moeten worden weggeschreven naar harddisk. Als de stroom uitvalt is er dus een probleem! Op zo'n moment dient alles te worden nagelopen en hersteld te worden [fsck] en dat kost tijd. De hele schijf moet worden doorlopen. Om dit probleem op te lossen zijn zogeheten journalling filesystems bedacht. Deze houden een journal [logboek] bij op de disk met de daarin de transacties die nog moeten worden doorgevoerd in de inhoudsopgave van het fs. Deze data wordt {waarmee dus de eigenlijke data op de disk wordt beschreven} wordt vaak metadata genoemd. Deze data wordt bij een ext3-bestandssysteem opgeslagen in een journal-bestand in de root-directory

**e2fsck** : check van e2fs systeem

**mke2fs**: het maken van een e2fs filesystem met de optie -t ext3 device wordt een ext3 gemaakt [= een verbeterde e2fs + journal]

**debugfs**: debuggen van het filesystem. Er wordt vervolgens een aantal opties gegeven [stats - open - features]

**dumpe2fs** : output van de parameters e2fs

**tune2fs**: een commando om parameters mee te geven: bv aangekoppelde keren van het systeem, errorchecking. Dit wordt gedaan met optie **-c** wanneer deze een waarde 0 krijgt worden de parameters niet uitgelezen.

### 104.3 Het koppelen van file-systemen.

In het linux systeem staat geconfigureerd welke partitie tijdens het opstarten

de / (root) van het systeem wordt. Dan worden er koppelingen gelegd tussen file-systemen en directories. **mount** zonder verdere opties en argumenten geeft een weergave van de op dit moment gemounte (gekoppelde) file-systemen. Op dit systeem valt dan te zien dat aan de **root /dev/hda2** gekoppeld is en dat daarop **reiserfs** als filesystem is geïnstalleerd. Het ontkoppelen van directories en/of partities (en het daarop staande file-systeem) kan met het "**umount**". Alleen als niet iemand anders gebruik maakt van de directory. Anders wordt de melding "device is busy" gegeven.

Tijdens het **opstarten** worden een of meer file-systemen **gemount**. Dit gebeurt door opstartscripts die allerlei controles doen en diverse systeembestanden lezen. Een van die belangrijke bestanden is **/etc/fstab**. In dit bestand staat een lijst van de beschikbare file-systemen en op welke plaats zij gemount moeten worden.

Elke regel bevat als eerste de device-file van de betreffende partitie, vervolgens de directory waarmee de koppeling tot stand moet worden gebracht, daarna het type file-systeem en daarna een aantal opties en vlaggen.

Men kan daarbij uit de volgende opties kiezen:

- defaults: geen speciale opties
- user: een normale gebruiker mag dit filesystem mounten. Als een gewone gebruiker het commando "mount /dev/fd0" wordt dit device gekoppeld.
- ro: read-only filesystem. Kan alleen gelezen worden.
- noauto: wordt niet automatisch gemount als het systeem start.
- exec: van dit filesystem mogen gebruikers (niet) programma's opstarten.

De twee vlaggen hebben de volgende betekenis. De eerste vlag betekent dat het dump programma van dit filesystem een back-up moeten maken (1=ja, 0 = nee). De tweede vlag heeft invloed op het controleren van het filesystem check. Een 0 betekent geen check, een 1 betekent als eerste checken en een 2 betekent overige. Voor het root filesystem staat er een 1, alle andere een 2. Het bestand **/etc/fstab** kan door u als systeembeheerder naar wens en met behulp van een editor als VI worden aangepast.

```
rene@linux:~> cat /etc/fstab (bv. mijn SUSE:ik heb dus meer mogelijkheden om aan te geven)
/dev/hda2      /          reiserfs acl,user_xattr   1 1
/dev/hda1      swap       swap          pri=42              0 0
devpts        /dev/pts   devpts        mode=0620,gid=5    0 0
proc          /proc      proc          defaults            0 0
usbfs        /proc/bus/usb  usbfs        noauto              0 0
sysfs        /sys       sysfs        noauto              0 0
/dev/cdrecorder /media/cdrecorder subfs        fs=cdfss,ro,procuid,nosuid,nodev,exec,iocharset=utf8 0 0
/dev/fd0      /media/floppy subfs        fs=floppyfss,procuid,nodev,nosuid, sync 0 0
```

**104.4 Quota.** Hoe voorkomt u dat een gebruiker de totale beschikbare ruimte van het /home filesystem in beslag neemt. De oplossing is het diskquota-systeem. Daarmee kan men een niet alleen een maximale grens instellen voor het aantal diskblokken dat een gebruiker mag consumeren, maar ook voor het aantal inodes. Het linux quota systeem werkt alleen voor het ext2 en wordt ingesteld per ext2-filesystem. Als u een redelijk recente kernel heeft zal de ondersteuning al binnen de kernel geregeld zijn.

Het toevoegen van de quota-optie kan in het bestand **ect/fstab** gebeuren. Wilt u controleren op userquota dan voegt u **usrquota** toe, wilt u controleren op groupsquota dan voegt u **grpquota** toe. De regel in **/ect/fstab** wordt dan bv.  
**/dev/hdc3 /home ext2 defaults,usrquota,grpquota 1 2**  
Daarnaast moeten er in de top-directory van de filesystemen die gecontroleerd moeten worden de bestanden **quota.user** en **quota.group** te staan. Deze mogen alleen eigendom van en lees- en schrijfbaar zijn voor de root.

```
Dus #touch /home/quota.user
    #touch /home/quota.grp
    #chmod 600 /home/quota.user
    #chmod 600 /home/quota.group
```

Nu zijn de juiste bestanden gemaakt en voorzien van de juiste rechten. Na een herstart checkt het systeem de nieuwe bestanden en voert controles uit.

Met **quotacheck** controleert de gewenste filesystemen en genereert aan de hand daarvan een quota-bestand. In het volgende voorbeeld start u handmatig om een reboot overbodig te maken (als root).

#### **quotacheck -avug**

De gebruikte opties hebben de volgende betekenis:

- a gebruik de in de fstab aanwezige filesystemen
- v geef informatie over wat er gebeurt
- u controleer de userquota
- g controleer de groupquota

De volgende stap is het aanzetten van de quotabewaking. Dit kan met het commando **quotaon**. **quotaoff** kan dus gebruikt worden om het weer uit te zetten.

Een van de meest gebruikte commando wordt dan **edquota**. U kunt met dit programma individuele quota van gebruikers instellen. Vervolgens wordt de editor gestart (standaard vi -afh. van de EDITOR environment variabele). U kunt nu de limieten aanpassen. Er is een softlimit, een hardlimit en een grace periode. Er wordt in het boek niet ingegaan of dit met een ext3 of een reiser anders gaat.

### **104.5 Het gebruik van toegangsrechten om de toegang tot bestanden mee te regelen**

Als gebruiker mag u bepaalde zaken wel doen en bepaalde zaken niet. Dit mechanisme beveiligt linux. Het bestaat onder meer uit de volgende onderdelen:

- elke bestand behoort toe aan een gebruiker. Het '**User-ID**' wordt opgeslagen in de eerste inode van het bepaalde bestand.
- elk bestand behoort ook toe aan een groep. Het '**Group-ID**' wordt eveneens opgeslagen in de eerste inode van het bepaalde bestand.
- elk bestand kent ook **toegangsrechten**. Deze bepalen wie wat mag doen. Er zijn drie onderdelen: de rechten van de **eigenaar**; de rechten van de **groep**; en de rechten van een **ieder ander**.
- commando's om de toegangsrechten voor gebruikers in te stellen en commando's om te wijzigen

**chmod**. Het meest gebruikte commando om de rechten van een bestand te wijzigen. Stel u heeft het volgende bestand in uw directory staan.

```
ls -l bestand: -rw-r--r-- 1 cursus users 104 Jul 13 15.49 bestand
```

U ziet hier als eerste informatie in de regel de toegangsrechten staan. Het eerste karakter is al eerder aan bod gekomen. De 'd' staat voor directory, de 'b' voor een blockdevice, de 'c' voor een character device. De - geeft veelal aan dat het een tekstbestand is. De resterende negen karakters bevat de toegangsrechten voor de verschillende drie soorten gebruikers (drie voor de user, drie voor de group, en tenslotte drie voor ieder ander -other-). Een **r** voor leesrechten (read), een **w** voor schrijven (write) en een **x** (execute) voor opstartrechten.

De rechten zijn te veranderen. **chmod u-rw+x bestand ; ls -l bestand** geeft:

```
---xr--r-- 1 cursus users 104 Jul 13 15.49 bestand
```

**chmod** zorgt er voor dat van de userrechten r/w (cursus) wordt ingetrokken en dat het execute-recht wordt toegekend. Zo kunt u ook de groeprechten (users) veranderen, bv.door **chmod g-r+wx bestand**. De rechten van ieder ander (other)met bv. **chmod o-r+x bestand**.

U kunt ook in een keer alles wijzigen. **chmod u=rw g=r o=r**.

U kunt het ook numeriek doen. Read = 4. Write = 2. Execute = 1

**chmod 600 bestand** geeft dus:

read en write voor de user  
niets voor de group  
niets voor de others

**chmod 755** bestand geeft dus:

read, write en execute voor de user (4+2+1=7)  
read en execute voor de group (4+1=5)  
read en execute voor others (4+1=5)

Bij **directories** geldt dit ook. Een **r** mag de inhoud van deze directory bekijken. De **w** mag bestanden aanmaken in deze directory. De **x** mag naar deze directory toe.

De **umask** wordt gebruikt om aan te geven welke rechten bestanden krijgen als u ze voor het eerst aanmaakt. Wordt ook wel 'user-file-creation mask' genoemd. Met de optie **-S** wordt het in symbolische vorm afgedrukt.

Op mijn SUSE machine. `umask -S` geeft `u=rwx g=rx o=rx`

Zonder optie geeft mask 022. Hier is het namelijk een optelling van de bits die uiteindelijk niet in de rechten komen mask 022 heeft dus als rechten 644. Dus hier gaat het een beetje fout - zie vraag. 4 bij mij geeft ie 0022

Er zijn een paar **speciale toegangsrechten** die worden gebruikt om bepaalde zaken mogelijk te maken, maar worden niet veel gebruikt. Het gaat om de rechten SUID, SGID en 'sticky bit'. Onoordeelkundig gebruik van deze rechten is een absolute bedreiging voor de veiligheid van uw systeem.

**chattr** is een speciaal commando dat aan bestand nieuwe/verborgen eigenschappen kan geven. Bv. `chattr -u /home/...` maakt dat het bestand niet verwijderd kan worden. optie `-c` [compress] pakt het bestand in

**SUID** of set user ID bit. Het regelt dat het effectieve user-id van een proces dat dit programma draait het user-id is van het programma en niet van de gebruiker die het start. Door het betreffende programma SUID recht te geven en de eigenaar van het programma root te maken worden de beveiligingen omzeilt. Tijdens de uitvoering van het programma is het UID van de root en niet van de gebruiker. Dit is tevens een gangbare methode om te proberen in een systeem in te breken. Het toekennen van het SUID recht kan door middel van `chmod u+s` of `chmod 4000`

**SGID** werkt hetzelfde als SUID, maar dan wordt het group-id van het betreffende programma gebruikt. Op sommige systemen betekent het dat bestanden die in een bepaalde directory worden geplaatst automatisch het group-id krijgen als de directory zelf. Toekennen kan met `chmod g+s` of `chmod 2000`

Als **Sticky Bit** wordt gezet wordt de tekstimage van een programma opgeslagen op de swapdevice voor een snellere toegang. De gebruiker kan geen bestanden verwijderen uit directories waarvan zij geen eigenaar zijn. Append-only directory. Met een `ls -l` comamndo ziet u een **t** achter de toegangsrechten. Een veel gebruikte toepassing is de /temp directory. Toekennen kan met het commando `chmod o+t` of `chmod 1000`.

**104.6 Het beheren van bestandseigendom.** Voor het wijzigen van de eigenaar en de groep bestaan er twee commando's. Het **chown** (change owner) commando wordt gebruikt om de eigenaar van een bestand te wijzigen. Dit kan uiteraard alleen de root.

Het **chgrp** commando wordt gebruikt om de groep waartoe een bestand behoort te wijzigen. Eenvoudig in gebruik:

`chown piet bestanden` (eigenaar is veranderd naar piet)  
`chgrp anderen bestand` (groep is veranderd naar anderen)

De commando's kennen allebei de optie **-R** waarmee recursief door een directory wordt gegaan. Alle bestanden in deze directory en sub-directories worden dan aangepast. Het chown commando kan ook worden gebruikt om een chgrp uit te voeren. chown piet.anderen bestand. Dit commando wijzigt zowel het user-ID als de group-ID.

#### 104.7 Het maken en wijzigen van harde en symbolische links.

Er zijn twee soorten links: **de symbolische en harde links**. Het begrip link hoeft niet tot verwarring te leiden als u weet dat een bestandsnaam los staat van een bestand.

```
rene@linux:~>
ln -s bestand bestand2 ; ls -l bestand
-rwxr-xr-x 1 rene users 104 Jul 13 15.49 bestand
lrwxrwxrwx 1 rene users 7 Jul 19 20.26 bestand2 -> bestand
```

Aan het begin staat een l van link. De link neemt veel minder ruimte in (7 ipv 104). De naam bestand2 verwijst naar bestand. **De optie -s in de opdrachtregel geeft aan dat een symbolische link is**. Een link heeft alle rechten omdat uiteindelijk de rechten van het doelbestand gelden.

Een harde link wordt met het zelfde commando (ln) gemaakt, maar is als we de uitvoer van een ls bekijken, minder makkelijk te herkennen.

```
rene@linux:~>
ln bestand bestand3 ; ls bestand* -il
751658 -rwxr-xr-x 2 rene users 104 Jul 13 15.49 bestand
751734 lrwxrwxrwx 1 rene users 7 Jul 19 20.26 bestand2 -> bestand
751658 -rwxr-xr-x 2 rene users 104 Jul 13 15.49 bestand3
```

U ziet dat het bestand3 gemaakt is met een harde link. Het is precies het zelfde bestand (grootte) en het bevindt zich op dezelfde inode van de harde schijf. Met de optie **-i** verschijnt het inode-nummer. Het gebruik van links kan erg gemakkelijk zijn. Als u in een /usr/local/bin directory PATH een link wil aanbrengen in een andere directory. Een symbolische link verdient de voorkeur. Het scheelt ruimte, maar het bestand verdwijnt niet als de symbolische link wordt verwijderd. Dit gebeurt wel bij een harde link.

Bij het kopiëren of op computertape vastleggen van hele directory structuren dient u er op te letten of de symbolische links mee worden gekopieerd. Uit man

```
cp
cp -d bewaar links
cp -s maak symbolische links vanaf de bestemming naar bronbestanden ipv de bestanden te kopiëren
cp -l maak harde links vanaf de bestemming naar de bronbestanden
```

#### 104.8 FHS en het plaatsen van bestanden in de juiste directory.

is een standaard die de indeling van linux-systemen beschrijft. Het bestaat uit drie delen.

- 1) De root directory,
- 2) de /usr hiërarchie
- 3) en de /var hiërarchie.

Het feit dat een bestand in het FHS wordt genoemd betekent niet dat dit bestand ook in elke distributie voorkomt of op deze plaats staat. Het betekent alleen dat de FHS voorstelt om het betreffende bestand op een bepaalde plaats neer te zetten. De inhoud van een **root-filesysteem** moet voldoende zijn om een systeem te starten, repareren, herstellen en terug te halen. Dit inclusief gereedschappen, configuratiebestanden, en opstartbestanden zoals de kernel en andere essentiële zaken.

De linux-directory structuur

**/ root directory**

**/bin** : essentiële commando's. Bevat programma's die in ieder geval zeker nodig zijn in single-user mode. Er mogen geen subdirectories zijn.

**/boot** : bootloader en kernel. Alles nodig voor het booten behalve de opstartscripts. Zij bevat bestanden die de kernel nodig heeft voordat user-mode programma's worden gestart.

**/dev** : device bestanden. Lokatie voor speciale en device bestanden

**/etc** : host-specifieke configuratie-bestanden. Bevat configuratie-bestanden. Er

horen hier geen programma's te staan. Er kunnen subdirectories zijn, bv van de configuratie van de x(11)-server of van etc/opt voor configuratie van optionele zaken (kde-gnome).

**/home** : home directory van gebruikers. Bevat individuele directories van gebruikers van een systeem. Op grotere systemen kan men gebruikers weer onderverdelen in groepen. Bv /home/studenten/individuele gebruiker.

**/lib** : essentiële gedeelde bibliotheken en kernelmodules die nodig zijn om het systeem te starten en de commando's van het root-filesysteem te kunnen verwerken.

**/mnt** : mountpoint voor koppeling (tijdelijke) filesystemen

**/opt** : optionele hardware pakketten. Is bedoeld voor add-ons en extra software-pakketten. Meerdere subdirectories (man-doc-include-lib)

**/root** : home-directory van de root gebruiker. Probeer deze zo klein mogelijk te houden. Het is beter er een goede gewoonte van te maken om het systeem als gewone gebruiker te benutten en indien nodig van het root-account gebruik te maken.

**/sbin** : essentiële systeem-programma's, maar minder noodzakelijk dan /bin

**/tmp** : tijdelijke bestanden. Men mag ervan uitgaan dat deze directory zonder grote problemen mag worden leeggemaakt tijdens het opstarten van het systeem.

**/usr** : tweede hiërarchie. Kan door meerdere systemen worden gedeeld. Is zelf weet onderverdeeld in subdirectories (X11R6, X386, bin, games, include, lib, local, sbin, share, src)

**/var** : variabele gegevensbestanden. Is voor opslag van variabele gegevens, inclusief printerspoolbestanden, administratieve en logging-in gegevens.

Subdirectories (kunnen) zijn account, cache,log, var, message, opt, run, spool, state, tmp).

**Bestanden zoeken.** Het programma **which** kunt u gebruiken als u wilt waar waar een bepaald commando staat. `which ls` geeft /bin/ls. Het bestand staat in de map /bin Het commando `type i` is ook te gebruiken. Met `type` wordt een alias ook weergegeven (`which` doet dat niet)

**find** wijkt wat af van de gebruikelijke manier van invoeren. De directory waarin gezocht moet worden. De naam van het gezochte bestand. En wat er mee moet gebeuren. Zo kan men bestanden vinden die voor een bepaalde datum zijn gemaakt, of niet groter dan een bepaald formaat zijn. Naam/tijd/grootte

```
rene@linux:~> find /home -size +100k zoekt alle bestanden die groter zijn dan 100K.
```

**Locate** Het find commando kan veel tijd nodig hebben en het systeem behoorlijk belasten. Als er nu 's nachts een database werd gemaakt met daarin alle bestandsnamen, dan zou u die overdag met een commando kunnen raadplegen. Het maken van een database die het **locate** commando gebruikt kan met het commando **updatedb**. Het resultaat wordt opgeslagen in var/lib.

**Slocate** security enhanced version of the GNU Locate. Soms is dit commando een alias van locate

**whereis** whereis - locate the binary, source, and manual page files for a command

```
etc/updatedb.conf: niet in SUSE
```

## 110: X-window (13)

Boek 2 Hoofdstuk 7 bladzijde 135 t/m 178

110.1 Installeren en configureren Xfree86 (5)

Boek 2 Hoofdstuk 8 bladzijde 180 t/m 195

110.2 Installeren van een display manager (3)

110.3 Dit leerdoel is vervallen

## 110.4 Installeren en configureren van een Window Manager omgeving (5)

### 110.1 Het X window systeem.

De **X-server** bindt zichzelf aan een muis, toetsenbord en grafische kaart en is verantwoordelijk voor alle grafische handelingen. Deze handelingen worden verricht als daar opdracht voor wordt ontvangen. De X server is netwerk georiënteerd wat betekent dat de opdracht van een andere computer af kunnen komen.

De **X-client** is een programma die X opdrachten verstuurt naar een X server toe. Dit kan een programma zijn zoals xterm.

**Window manager** De eerste twee componenten zijn genoeg voor een werkend systeem, maar men wil vaak meer. De randen rondom een venster, de menubalk enz. De windowmanager is zelf ook een x-client [**kde/gnome**].

Er bestaan een groot aantal X servers in de linuxwereld Xfree86 en nu is X.org in opkomst. Na de installatie volgt het moeilijkste deel: de configuratie. Voorafgaand zijn een aantal zaken erg belangrijk. U moet tot in detail kunnen nagaan en later kunnen opgeven wat het horizontale en verticale bereik is van uw monitor, het type grafische kaart, de gebruikte chip op uw grafische kaart, hoeveelheid videogeheugen, het type muis en het gebruikte model toetsenbord.

**xfconfig86** is een sober script met veel tekst en uitleg over de configuratie van de X server. Er worden verscheiden vragen gesteld waarop u dient aan te geven wat eea moet zijn. En er worden nogal wat vragen gesteld. Van welk model muis u heeft tot de frequenties van uw monitor. Het te hoog aangeven van deze waarden kan schadelijk zijn voor uw hardware: let dus op! Control-Alt-Backspace brengt de X server down, niet linux zelf.

**XF86Setup** is al wat gebruikersvriendelijker. Hier kan het grafisch het hele gebeuren worden ingesteld.

Uiteindelijk moet het resulteren in een bestand in **/etc/X11/XF86Config** waarin alle instellingen zijn opgeslagen en met het starten van de X server dit bestand wordt ingelezen.

**xvidtune**: dit programma kan worden gebruikt om de instellingen van de Xwindow server aan te brengen. Het is in staat om softwarematig het weergegeven scherm horizontaal en verticaal te versmallen en te verbreden. U start het programma zonder argumenten op. Ook kan het gebruikt worden met enkele opties, zoals de optie **-show**, die de op dit moment actieve moderegeling van het configuratiebestand **xf86config** laat zien.

Een van de zaken waarin X voorziet is de mogelijkheid om applicaties naar eigen wens aan te passen. Denk daarbij aan zaken als achtergrondkleur, venster, titel etc. Dit worden **programmaresources** genoemd. Zo is er het **usr/X11R6/lib/X11/X-resources** bestand met een groot aantal algemene instellingen. Tijdens het opstarten van X, of het aanmelden van een gebruiker, wordt dit bestand met het **xrdb** commando ingelezen. Vervolgens kunnen andere lokale gebruikersbestanden bv **~/.Xresources** worden toegevoegd met het **xrdb -merge** commando. **xrdb** is dus een database gereedschap voor het invoegen van resources voor X.

### 110.2 Het installeren van een displaymanager

Er zijn een groot aantal window-managers beschikbaar voor Linux. Sommige gaan daarvan verder dan een windowmanager alleen en leveren een hele desktop omgeving zoals KDE of GNOME.

**KDM**: Hoewel de indruk wordt gewekt zijn de directories **etc/x11/kdm** en **etc/x11/gdm** niet vanzelfsprekend. Configuratie gebeurt met de standaard XDM-

bestanden en daarnaast met een bestand `kdmrc`, dat afhankelijk van de distributie op diverse plaatsen kan staan. **KDE**: configuratiebestanden staan in `opt/kde/share/config [suse]`

De displaymanager **GDM** is uit het gnome project voortgekomen. **GNOME**: de configuratiebestanden staan in `opt/gnome/share/gnome [Suse]`

#### 110.4 het installeren en configureren van een Window Manager omgeving

Het voert te ver om hier de configuratiebestanden voor alle beschikbare window managers te gaan behandelen. Toch zijn er wel een aantal zaken die overall hetzelfde zijn. Dat zijn de afzonderlijke configuratiebestanden die worden gebruikt **voordat** de window-manager wordt gestart: **Xsession**, **.xinitrc** en **.Xdefaults**.

**Xsession** bestand in de XDM directory bevat het script dat nu het succesvol inloggen dient te worden uitgevoerd. Hier kan men bv commando's inzetten voor het opstarten van een windowmanager voor elke afzonderlijke gebruiker

**.xinitrc** Dit bestand wordt meestal uit de `etc/skel` directory in de home directory van elke nieuwe gebruiker geplaatst. Hierin kunnen individuele zaken worden geregeld zoals het starten van een window-manager voor elke afzonderlijke gebruiker.

**.Xdefaults** Men kan door middel van de environment-variabele `$ENVIRONMENT` de configuratie van X beïnvloeden. Deze verwijst dan naar een X resource bestand Als deze variabele niet bestaat zal het bestand `$HOME?.Xdefaults-hostname` gebruikt worden.

**X terminals**. Het XDM programma kan ook worden gebruikt door zogenaamde X terminals. Dat zijn grafische terminals zonder een eigen harddisk die een verbinding kunnen maken via het eigen netwerk met een X server. Als u een X-terminaal wilt toestaan om in te loggen op deze X server vermeldt u de betreffende X terminaal in het `Xaccess` bestand [in `etc/X11/xdm/Xaccess`]. Behalve de mogelijkheid om andere systemen toegang te geven tot uw systeem kunt u ook de **CHOOSEER** gebruiken. Dat is een klein venster waarmee andere systemen eerst de gewenste server kunnen kiezen voordat ze daarop proberen in te loggen. Als u de **CHOOSEER BROADCAST** optie functie gebruikt wordt door `xdm` eerst in het netwerk gevraagd wie er allemaal een XDM venster kan leveren en wordt vervolgens de lijst automatisch samengesteld.

**Xsession**. Nadat men succesvol is ingelogd wordt als eerste het `Xsession` script uitgevoerd. Hierin kan men allerlei zaken regelen voor de individuele gebruiker zoals het opstarten van iemands favoriete window manager, e.d.

**X terminal**. Onder `x` kan een terminaal worden gerealiseerd door het opstarten van een X terminaal programma. U dient dit niet verwarren met de bovenstaande X terminals als hardware [vaak zonder harde schijf]:  
`xterm`: de klassieke X-terminaal. Het kent een groot aantal opties  
`aterm`: onderdeel van `afterstep`. Kent ook veel opties  
`konsole`: de standaard van KDE. Een van de `xterm` programma's met een eigen menubalk. Daarin kan men de gewenste shell selecteren of een veel gebruikt programma selecteren.

**xhost**. Dit programma wordt gebruikt om andere gebruikers en/of systemen toegang te geven tot uw grafische sessie. Na dit commando mag elke gebruiker grafische weergaven lanceren op uw X server [ook als u hier minder blij mee bent]  
Het eerste display op het scherm is `:0`

**DISPLAY** environment variabele: door het instellen en exporteren van de env-variabele kan de weergave van een applicatie gestuurd worden.

**Afhankelijkheden**. Zoals in het vorige hoofdstuk is behandeld zijn X applicaties

afhankelijkheid van bepaalde programmabibliotheken. Met het **ldd** commando kunnen deze afhankelijkheden zichtbaar worden gemaakt. Bv **ldd usr/bin/xterm** geeft een aantal afhankelijkheden aan die aanwezig dienen te zijn.