

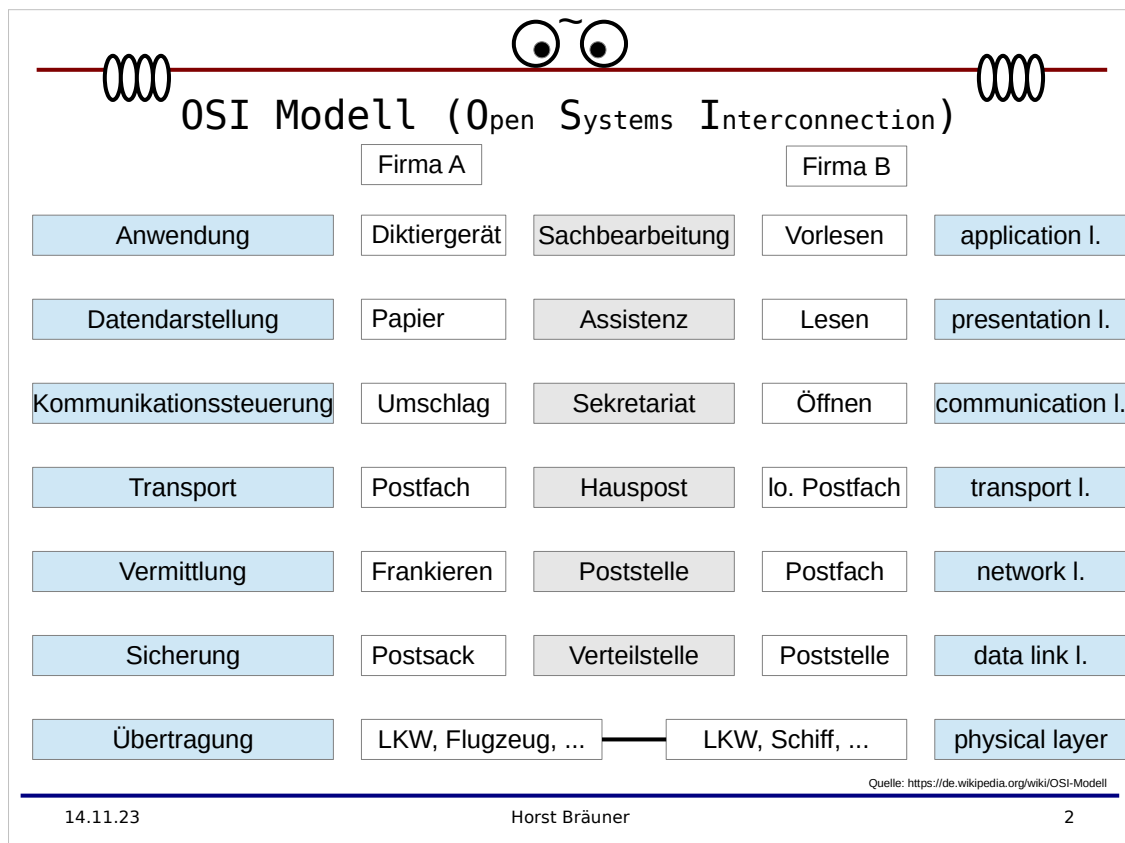
Praktische Datenverarbeitung 2022/23 - Grundlagen -

Aufwand etwa 4-5 UE

Inhalt

- OSI
- TCP/IP
- DNS
- URL
- Linux Eigenschaften, Datei- und Benutzerrechte, Konfigurationsdateien,
...
- Setup für die praktische Anwendung

Das ist ein grober Überblick. Die Grundlagen sollen alle Studierenden auf den selben Stand in diesen Themen bringen. In den weiteren Kapiteln werden diese Grundlagen vorausgesetzt.

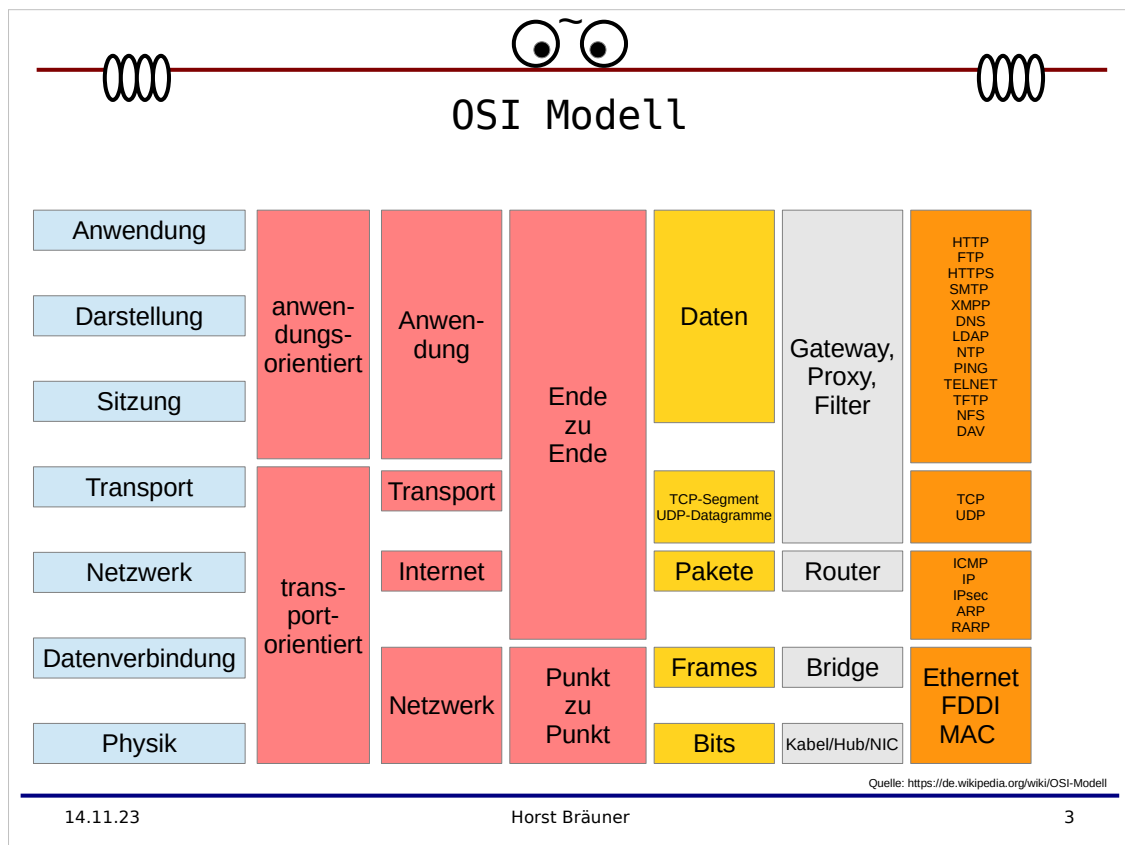


Zum Einstieg die bildhafte Darstellung des OSI-Modells zur Rechnerkommunikation. Ich gehe davon aus, dass das OSI-Modell den meisten bereits bekannt ist.

Die Kommunikation verläuft von links oben nach unten, nach rechts unten und nach rechts oben.

Wichtige Punkte:

- auf dem physical Layer muss zur erfolgreichen Übergabe an den anderen Kommunikationspartner dasselbe Medium („LKW“) verwendet werden. Die übrigen Schichten können auf jeder Seite anders ausgeprägt sein.
- im senkrechten Verlauf ist die Übergabe zwischen den Schichten definiert.



14.11.23

Horst Bräuner

3


Übersetzt in die Praxis ist die Abgrenzung nach OSI-Schichten weniger deutlich. Damit Sie die gängigen Begriffe einordnen können hier in alternativer Darstellung.

Zum Beispiel spricht man bei einer „Punkt zu Punkt Verbindung“ von den Ebenen 1 und 2. „1:n“ Verbindungen auf diesen Ebenen werden über eine Bridge (und die zugehörigen phys. Verbindungen) hergestellt.

Das OSI-Modell ist für die praktische Arbeit meist nur bis zum Netzwerklayer anwendbar.

TCP und UDP unterscheiden sich darin, dass TCP verbindungsorientiert ist und UDP verbindungslos. TCP versendet (Netzwerk-)Pakete und wartet auf eine „Antwort“ des Empfängers. UDP versendet (Netzwerk-)Pakete ohne auf eine „Antwort“ zu warten.

Bei den Protokollen wie HTTP, DNS, LDAP usw. ist eine strikte Zuordnung nach OSI-Schichten nahezu unmöglich.



TCP/IP Adressierung (IPv4)

32 Bit in 4 * 8 Bit durch „.“ getrennt, Sonderfall „localhost“ 127.0.0.1

0.0.0.0	00000000	00000000	00000000
255.255.255.255	11111111	11111111	11111111

Klasse A	Netzwerk-Kennung	Host-Kennung
Klasse B	Netzwerk-Kennung	Host-Kennung
Klasse C	Netzwerk-Kennung	Host-Kennung

private Adressbereiche			
Netzadressbereich	alt. Schreibweise	Anzahl Adressen	Anzahl Netze
10.0.0.0 10.255.255.255	10.0.0.0/8	$2^{24} = 16.777.216$	Klasse A, 1 Netz, 16.777.216 Adressen
172.16.0.0 172.31.255.255	172.16.0.0/12	$2^{20} = 1.048.576$	Klasse B, 16 Netze, 65.536 Adressen / Netz
192.168.0.0 192.168.255.255	192.168.0.0/16	$2^{16} = 65.536$	Klasse C, 256 Netze, 256 Adressen / Netz

14.11.23
Horst Bräuner
4

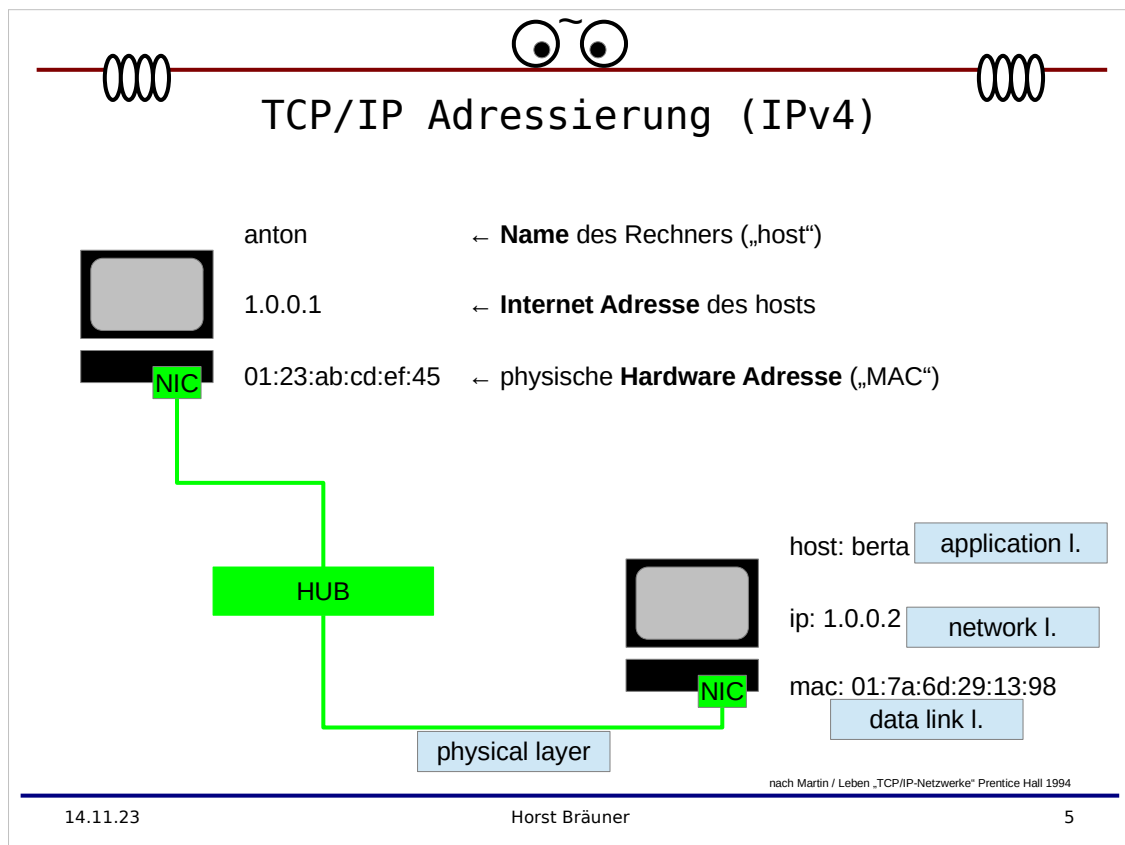
IP-Adressen

IP-Adressen haben eine Netzwerk- und eine Host-Kennung. Je nach Klasse können viele Rechner in wenigen großen Netzen oder wenige Rechner in vielen kleineren Netzwerken verwaltet werden.

IP-Adressen werden durch zentrale Stellen, z.B. über Internet-Provider vergeben. Es darf im öffentlichen Internet keine doppelten Adressen geben.

Für die Verwendung innerhalb einer Firma oder Organisation sind private Adressbereiche vorgesehen. Der private Adressbereich kann von jedem verwendet werden. Adressen aus dem privaten Adressbereich werden nicht über das Internet verbreitet („geroutet“). Wenn Rechner aus dem privaten Adressbereich mit dem Internet kommunizieren, werden die privaten Adressen mindestens über Router in öffentliche Adressen „übersetzt“.

Eine Ausnahme ist die 127.0.0.1. Diese Adresse verwendet jeder Rechner nur für sich selbst („localhost“). Über diese IP-Adresse kann der Rechner mit sich selbst über den Netzwerk-Layer kommunizieren.



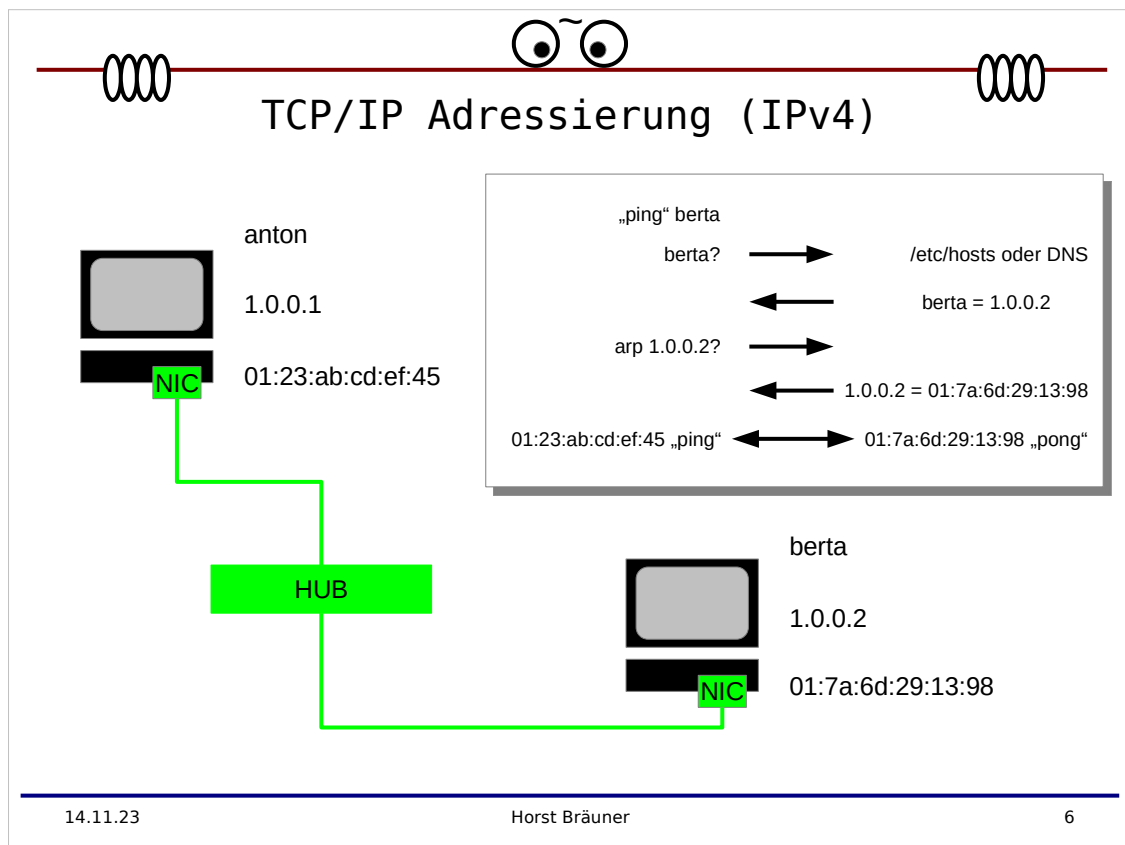
Am Beispiel zweier Rechner die typischen Kommunikationskanäle nach OSI-Layern.

Der phys. Layer verbindet die Rechner. Das kann z.B. ein Netzkabel sein, eine Funkverbindung oder theoretisch jedes beliebige Medium, auf das beide Seiten Zugriff haben.

Die Verbindung zum phys. Layer erfolgt über eine „Media Access Control“-Adresse (MAC). Diese ist für jeden Rechner in einem Netz eindeutig und einmalig.

Die IP-Adresse „übersetzt“ die MAC-Adresse in eine lesbare Form, die für die Kommunikation flexibel verwendbar ist. Sie kann im Bereich der privaten Netzwerke selbst vergeben werden.

Der Name eines Rechners ist über „DNS“ (Domain Name Service) bzw. „hosts“ (hosts-Datei) und damit im Anwendungslayer definiert.



Will ein Rechner mit einem anderen kommunizieren, erfolgt die Verbindung in mehreren Schritten.

Hier am Beispiel „ping“: Ping ist ein Programm, das das ICMP-Protokoll (Internet Control Message Protocol) nutzt um eine kleine Menge an Daten an einen anderen Rechner zu senden und auf ein „pong“ des anderen Rechners zu warten (exakt: icmp echo-request - icmp echo-reply).

Erster Schritt: DNS-Anfrage per Rundruf ins Netz

Erste Frage: „wer ist berta?“ bzw. „gibt es eine IP-Adresse für berta?“

Erste Antwort: „berta hat die IP Adresse 1.0.0.2“

ZweiterSchritt: ARP-Anfrage (address resolution protocol)per Rundruf ins Netz

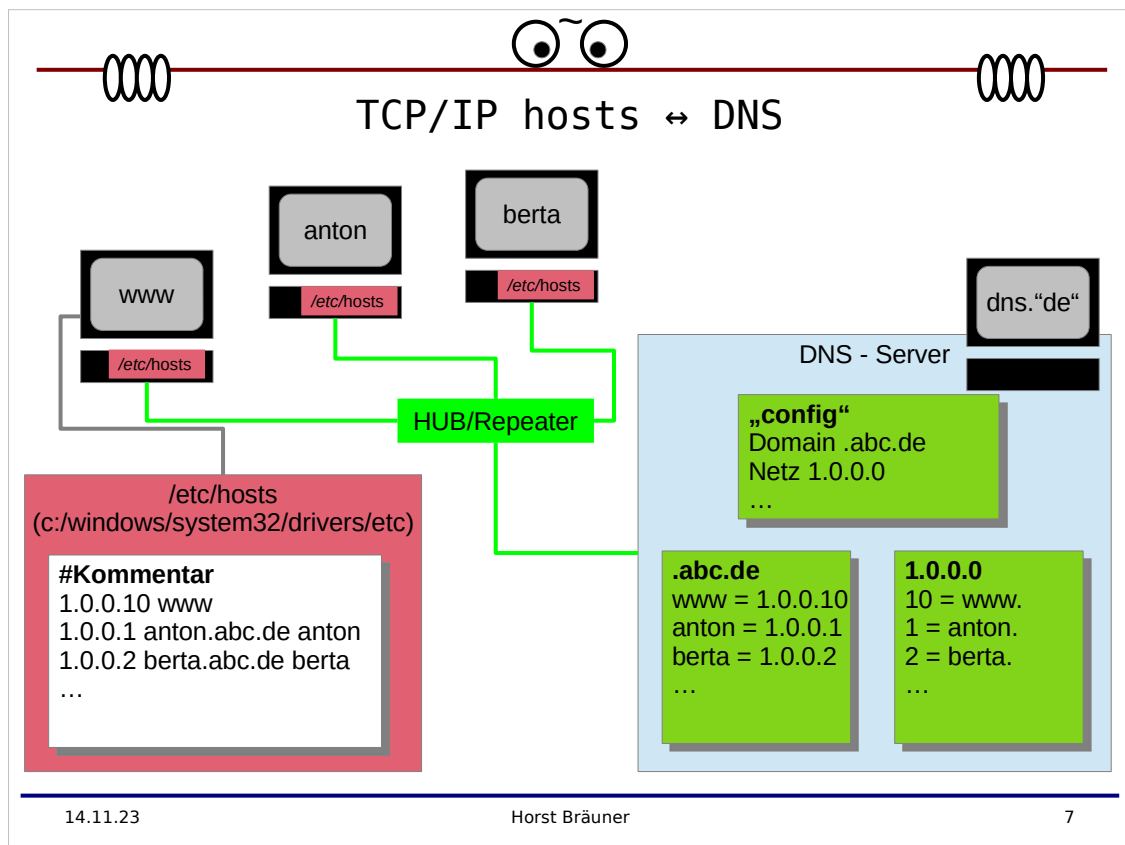
Zweite Frage: „gibt es eine physikalische Adresse von berta?“ bzw. „auf welchem physikalischen Weg erreiche ich berta?“

Zweite Antwort: „für Daten an berta, sende die Daten an 01:7a:6d:29:13:98“

Dritter Schritt: Datenpaket an 01:7a:6d:29:13:98

Dritte Frage: „hast Du (= 01:7a:6d:29:13:98) meine Daten erhalten?“ PING

Dritte Antwort: „ja, hier ist meine Antwort/sind meine Daten“ PONG

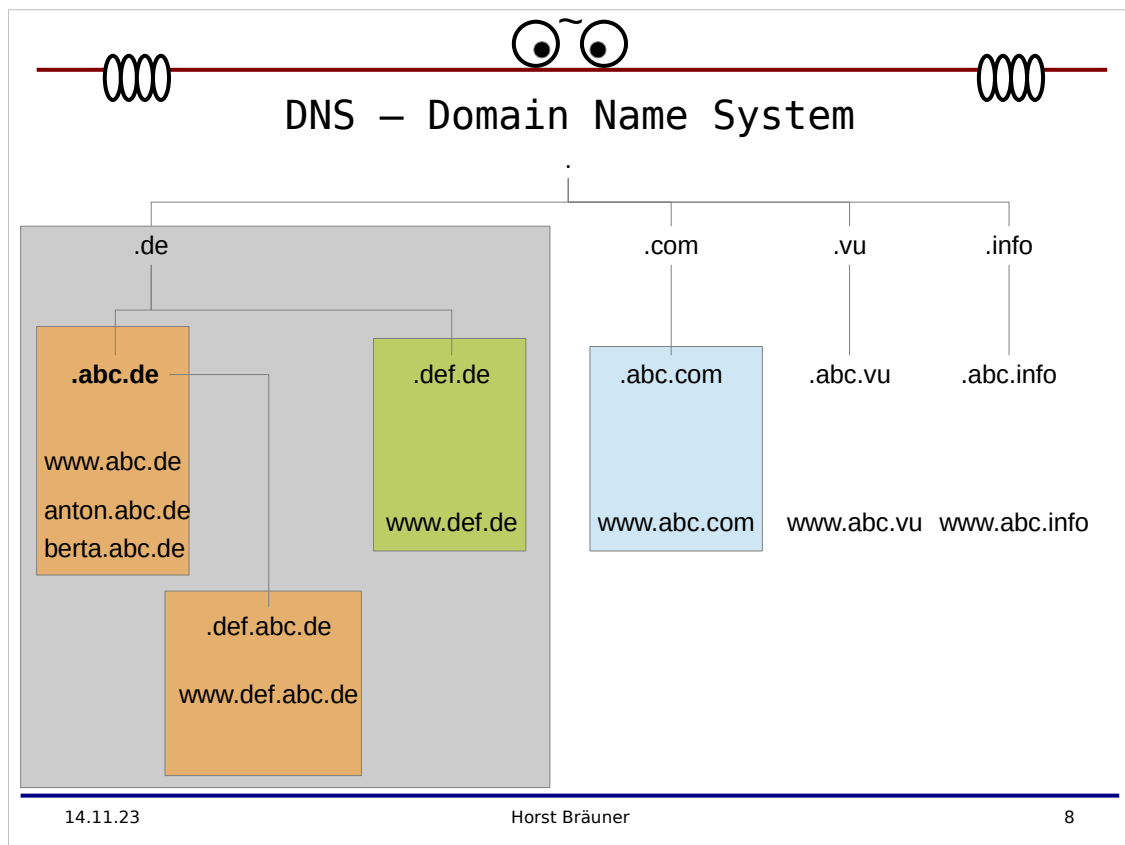


Die bereits Alternativen „hosts“ und DNS im Vergleich

Zu Beginn der Computer-Ära dachte keiner an eine massenhafte Verwendung. Namen von Rechnern gegen IP-Adressen aufzulösen erfolgte daher mit Listen, die auf jedem Rechner gepflegt wurden. Diese Listen finden heute noch Verwendung. Es ist innerhalb eines lokalen, begrenzten Netzwerkes eine schnelle und einfache Methode Namen IP-Adressen zuzuordnen. Zudem reduziert diese Methode das Datenaufkommen im Netz, da DNS-Anfragen entfallen. Der Nachteil ist die Verteilung der Listen auf alle Kommunikationspartner.

Das Domain Name System, DNS, wurde später entwickelt und wird im folgenden dargestellt.

Die Konfigurationen zur „hosts“-Methode finden Sie auf Unix/Linux-Systemen in der Datei /etc/hosts und auf Microsoft-Windows-Systemen unter c:/windows/system32/drivers/etc/hosts.




DNS arbeitet nach dem einfachen Prinzip der Delegation

Oberste Hierarchie ist der „.“. Darunter werden Verwalter für jede DNS-„Domäne“ eingesetzt, z.B. DENIC (www.denic.de) für die .de-Domäne. Meist gibt es für jedes Land ein eigenes „Network Information Center“ (NIC), das die Länder-spezifische Top-Level-Domain verwaltet. Die weiteren Top-Level-Domains wie .biz, .org, .name ... sind zum Teil frei zugänglich, siehe auch <https://de.wikipedia.org/wiki/Top-Level-Domain>. Jeder Domain Verwalter kann unterhalb seiner Domain weitere Sub-Domains erstellen und die Verwaltung für diese Domains delegieren.

Im obigen Beispiel sind 4 Top-Level-Domains (TLD) abgebildet, 5 Sub-Domains und 1 „Sub-Sub“-Domain. In jeder Domain befindet sich ein Rechner mit Namen „www“, zusätzlich in der Domain .abc.de die Rechner „anton“ und „berta“.

Der Vorteil des DNS ist, dass Anfragen innerhalb einer Domäne nicht über das gesamte Netzwerk erfolgen, sondern im Idealfall nur bis zur nächst höheren „Abzweigung“.

Wenn also www.def.abc.de nach anton.abc.de fragt, wird die Anfrage von .abc.de bereits beantwortet. Wenn www.def.abc.de nach www.def.de fragt liefert .de die Antwort. Erst wenn www.def.abc.de nach www.abc.info anfragt, eskaliert die Anfrage auf die oberste Ebene.

	
URL - Uniformed Ressource Locator	
scheme : // user @ host . domain : port / url-path ? query	
http://www.abc.de	Standard web-server mit default Seite („index.html“)
https://www.abc.de/irgendwo/eine/datei.php?inhalt=wichtig	Verschlüsselter Zugriff mit Abfrage an Applikationsserver
ftp://user:password@www.abc.de:8080/irgendwo/eine/datei.xyz	ftp-server mit user/passwort und eigenem Port
file:///tmp/datei.txt	Dateizugriff auf lokalem Rechner „Unix“
file:///c:/temp/datei.txt	Dateizugriff auf lokalem Rechner „Windows“
network:// smb://COMPUTER/freigabe dav://COMPUTER/freigabe	Netzwerk, z.B. smb, nfs, webdav, ...
mailto:me@myhost.com	E-Mail
14.11.23	Horst Bräuner
	9

Jede Information kann über eine URL (Uniformed Ressource Locator) beschrieben werden.

Zwingend erforderlich ist im Grunde nur das Protokoll und dessen Trennung. Die anderen Bestandteile ergeben sich aus der Ressource.

scheme bezeichnet das „Protokoll“ bzw. den Typ der Information (=Ressource)

:// Trennung des Protokolls

user ist der Benutzername oder der Benutzername und das Passwort

@ ist die Trennung zwischen user und host, falls user angegeben wird

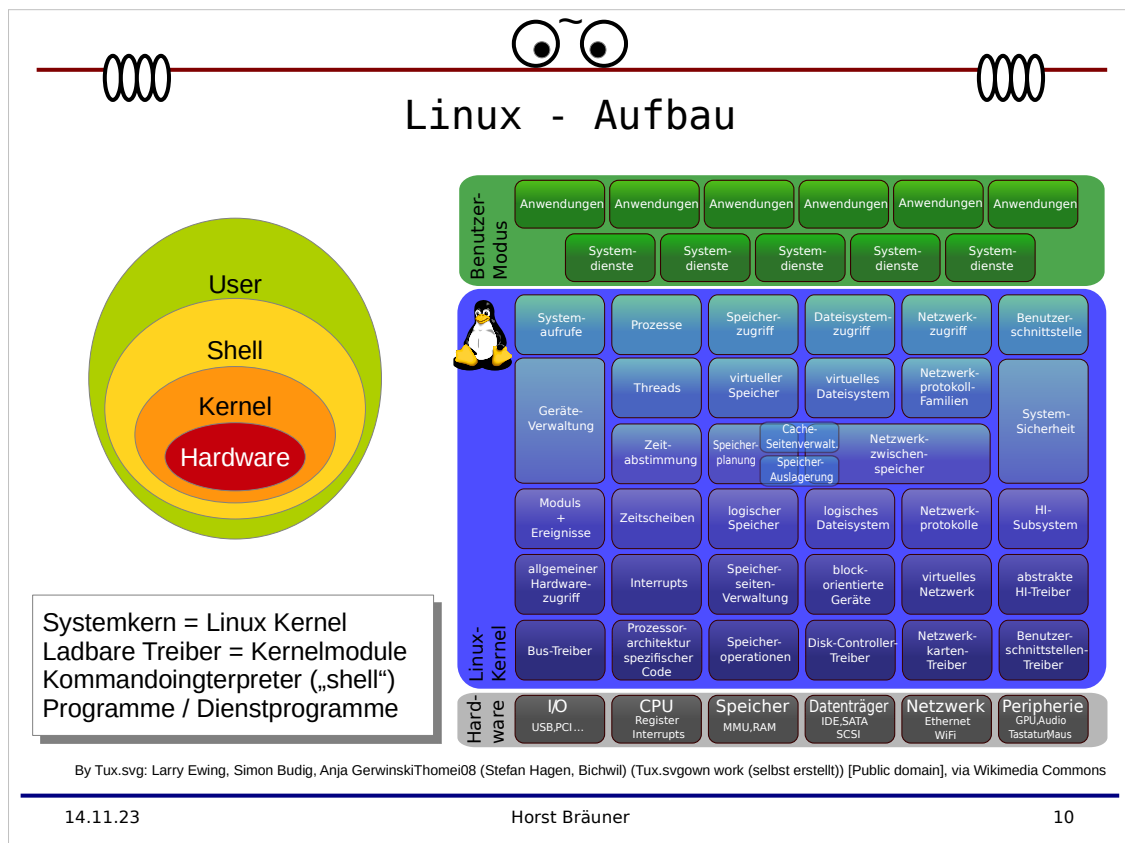
host ist der Name des Rechners, der die Ressource bereit hält

. ist die Trennung zur Domain (s. o. DNS), falls die Domain angegeben wird

: ist die Trennung zur Portnummer (), falls optional der Port angegeben wird

/ ist die Trennung zum Pfad zur Ressource

? ist eine mögliche Abfrage



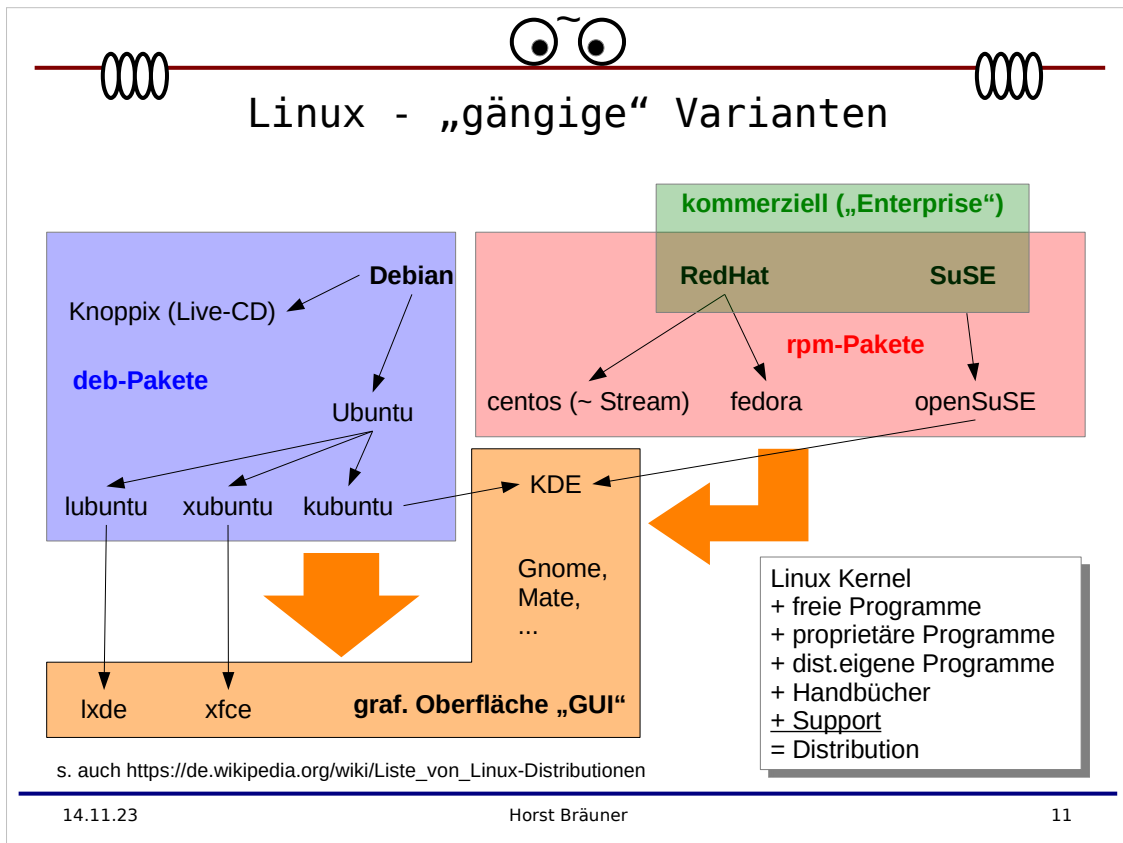
Der grundsätzliche Aufbau eines Linux Systems. Die Details sind hier nicht wichtig.

Linux besteht aus einem Systemkern, der mit der Hardware kommuniziert.

Um den Kern ist eine Shell, mit der mit dem Kern kommuniziert wird.

Um die Shell können dann weitere Komponenten/Dienste angesiedelt werden, die mit dem Benutzer interagieren.

Dieser Aufbau erlaubt es, dass Linux-Systeme in vielerlei Hardware verwendet werden kann; vom Dampfstrahler bis zum Kassensystem, vom Staubsauger bis zur Bierbrauanlage und vom PKW bis zum Desktop und Hochleistungsserver. Je nach Anwendung ist oft nur der Kernel und eine minimale Shell notwendig. Ein Staubsauger kann auf ein Display mit Maussteuerung verzichten ;-)




Linux Betriebssysteme werden grob nach dem Typ ihres Paketmanagements eingeteilt.

Eine Paketverwaltung (englisch package management software) ermöglicht die komfortable Verwaltung von Software, die in Programmpaketform vorliegt, auf einem Betriebssystem. Dazu gehören das Installieren, Aktualisieren und Deinstallieren (<https://de.wikipedia.org/wiki/Paketverwaltung>).

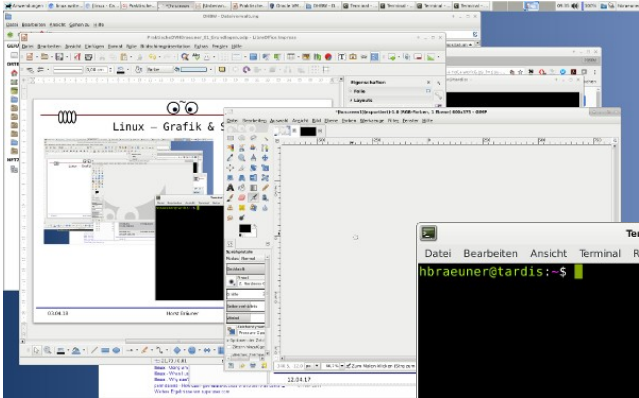
Die Bedienung und Funktion der jeweiligen Linux-Distributionen ist identisch. D.h. ein installiertes RedHat-Linux hat dieselbe Funktionalität und wird genauso angewandt wie ein Xubuntu.

Die Linuxe der Debian Gruppe können noch nach den jeweiligen grafischen Oberflächen unterschieden werden.

Im obigen Beispiel enthält der grüne Bereich die kommerziell unterstützten Linuxe, der blaue und rote Block die freien Varianten. Inhaltlich unterscheiden die Varianten sich nicht. Bei den kommerziell unterstützten Varianten ist jedoch ein Supportvertrag zwingend. Der graue Bereich enthält eine paar übliche, optionale grafische Oberflächen. Grafische Oberflächen werden später noch erläutert.



Linux – Grafik & Shell



„KlickiBunti“
= „Desktop“
Unity, xfce, lxde, kde,
gnome, ice, cinnamon,
nextstep...

„shell“
= Kommandozeile
sh, csh, ksh, bash, ...

```
Terminal - hbraeuner@tardis: ~  
Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
hbraeuner@tardis:~$
```

14.11.23

Horst Bräuner

12




Linux Systeme sind sowohl Server als auch Desktop-System und können auch gleichzeitig als Server und Desktop fungieren.

Linux-Systeme benötigen keine grafische Oberfläche.

Für Linux-Systeme gibt es eine Vielzahl an grafischen Oberflächen. Sie unterscheiden sich in Optik und Bedienung und vor allem im Ressourcenverbrauch. Selbst bei sehr alter Hardware kann mit Linux ein grafisches System installiert und betrieben werden.

Wir gehen in dem Kapitel X-Windows noch auf verschiedene Beispiele dazu ein.

Die meisten Administratoren verwenden die „Shell“ zur Bedienung ihrer Systeme.



Linux - Grundlagen

- alles ist eine Datei
 - jede Komponente (auch Hardware, Prozesse) wird wie eine Datei behandelt
- Datei- und Benutzerrechte
 - in Kombination mit ACL („Zugriffs“-Listen) werden komplexe Rechte-Systeme konfiguriert
- Keine Laufwerksbuchstaben
 - hierarchischer Verzeichnisbaum mit beliebig eingehängten (ge „mount“eten Verzeichnissen)
- Konfiguration in Textdateien
 - jederzeit „von Hand“ editierbar, übertragbar, automatisierbar
- Befehlszeile und Scripte
 - grundsätzlich ist alles automatisierbar
- Case Sensitivity
 - diese [DATEi] ist nicht diese [daTei]

14.11.23

Horst Bräuner


13



Ein paar Grundsätze ...

da alles eine Datei ist, können auch alle Methoden, die auf Dateien angewendet werden auf alle Komponenten angewendet werden. Beispielsweise kann auch eine Netzwerkkarte kopiert, gelöscht und editiert werden. Ebenso kann eine Soundkarte kopiert werden. Dann haben Sie 2 Soundkarten. Ob das immer sinnvoll ist, ist nicht die Frage ;-) Sie können Prozesse löschen, verschieben, kopieren ... und Rechte gibt es für alle Komponenten nach demselben Prinzip. Das heißt, wenn Sie einem Anwender das Recht an der Soundkarte nehmen, hört dieser keinen Ton mehr.

Die Konfiguration über Textdateien ist ein wesentlicher Vorteil von Linuxen. Haben Sie eine Konfiguration erfolgreich getestet, steht sie Ihnen überall zur Verfügung. Deshalb ist auch grundsätzlich alles automatisierbar. Alles was Sie an einem System tun, können Sie automatisch identisch an einem anderen System ablaufen lassen.

Case-Sensitivity ist oft eine Umgewöhnung für Anwender proprietärer Systeme.



Linux – alles ist eine Datei (1)

Auszug aus /dev:

brw-rw----	1 root disk	8, 0	Apr 6 14:39	sda
brw-rw----	1 root disk	8, 1	Apr 6 14:39	sda1
brw-rw----	1 root disk	8, 2	Apr 6 14:39	sda2
brw-rw----	1 root disk	8, 5	Apr 6 14:39	sda5
lrwxrwxrwx	1 root root	8	Apr 6 14:39	shm -> /run/shm
crw-----	1 root root	10, 231	Apr 6 14:39	snapshot
drwxr-xr-x	3 root root	260	Apr 6 14:39	snd
brw-rw----+	1 root cdrom	11, 0	Apr 6 14:39	sr0
lrwxrwxrwx	1 root root	15	Apr 6 14:39	stderr -> /proc/self/fd/2
lrwxrwxrwx	1 root root	15	Apr 6 14:39	stdin -> /proc/self/fd/0
lrwxrwxrwx	1 root root	15	Apr 6 14:39	stdout -> /proc/self/fd/1
crw-rw-rw-	1 root tty	5, 0	Apr 6 14:39	tty
crw--w----	1 root tty	4, 0	Apr 6 14:39	tty0
crw-rw----	1 root tty	4, 1	Apr 6 14:39	tty1

attributes	owner/group	date	name
------------	-------------	------	------

attributes: b = block, c = character, l = link, d = directory, ...
sda... = Festplatten, sr0 = CDRom, snd = Soundkarte, tty... = „Terminal“

14.11.23
Horst Bräuner
14

Ein Listing aus einem Linux-System zur Übersicht und ersten Orientierung.


Sie sehen „sda“ als blockorientiertes Geräte Festplatte und deren Partitionen als sda1, sda2, sda5.

Sie sehen die Soundkarte „snd“ als Verzeichnis („d“ für „directory“ an erster Stelle) unter dem dann die einzelnen Bestandteile wie Sequenzer, Timer, Chipsatz usw. als Dateien abgelegt sind.

Sie sehen die Charakter-Geräte „tty“ als Repräsentanz verschiedener „Bildschirme“. TTY (vom englischen teletypewriter) steht als Abkürzung für Telex / Fernschreiber.

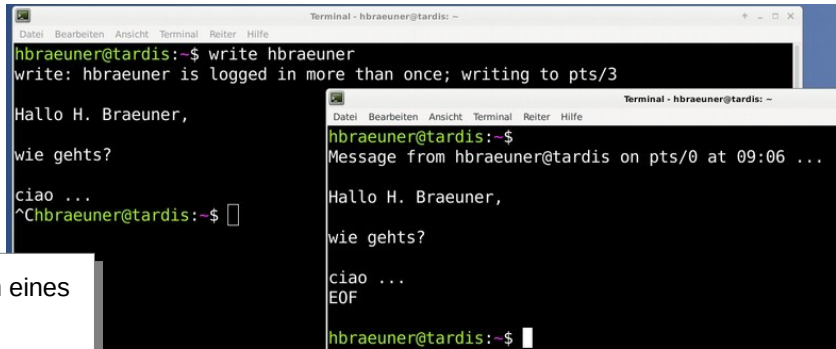
~

Linux – Alles ist eine Datei (2)



```
hbraeuner@tardis:~$ dd if=/dev/sr0 of=meinecd.iso
56880+0 Datensätze ein
56880+0 Datensätze aus
29122560 bytes (29 MB, 28 MiB) copied, 31,4504 s, 926 kB/s
hbraeuner@tardis:~$
```

CD/DVD in Datei kopieren




Text an Bildschirm eines users schreiben



14.11.23 Horst Bräuner 15

Beispiele...

Im obigen Beispiel sehen Sie, wie eine CD in eine .iso-Datei kopiert wird. Sie nehmen das Gerät `/dev/sr0` und kopieren dieses in eine Datei. `sr0` ist das CD-Laufwerk, `dd` ist der Befehl „duplicate device“.

Darunter ein Beispiel für einen primitiven „Chat“. Sie nehmen „write“ und einen Benutzer und „schreiben“ an dessen Bildschirm.



Linux – Datei- und Benutzerrechte (1)

```

horst@deeptthought:~$ ls -l
insgesamt 108
d rwx r-x r-x  2  heidi    gntms    4096   Mär 17 07:16   Arbeitsfläche
d rwx r-x r-x  2  hugo     checkers 4096   Dez 19 17:11  Bilder
- rw- rw- r--  1  hugo     checkers 948    Feb 14 13:42  cddb.toc
- rw- rw- r--  1  hugo     checkers 24320  Mär 17 17:15  DHBWScript.odt
d rwx r-x r-x  4  hugo     checkers 4096   Mär 29 16:02  Dokumente
d rwx r-x r-x  2  hugo     checkers 4096   Apr  2 14:29  Downloads
d rwx r-x r-x  2  helga    gntms    4096   Dez 30 17:24  dwhelper
d rwx r-x r-x 22  hugo     checkers 4096   Mär  3 17:17  Musik
- rwx rwx rwx  1  jens     users    28052  Feb 10 16:43  MVSarah.odt
d rwx r-x r-x  2  hugo     checkers 4096   Dez 19 17:11 Öffentlich
  
```

attributes	owner	group	size	last mod. date	name of file/directory
d/s/l/...	r w x	r w x	r w x		
type	user	group	other		

Typen: d = directory, l = link, s = special file, ...
Rechte: read, write, execute; Numerik: r = 4, w = 2, x = 1

14.11.23
Horst Bräuner
16

Rechte an Dateien (und anderen Komponenten) werden immer nach demselben Schema vergeben.

Sie haben 3 mögliche Rechte an einer Datei/Komponente:
 Lesen, Schreiben (enthält Löschen), Ausführen (bedeutet bei Verzeichnissen, dass Sie hinein wechseln dürfen)

Jedes Recht hat einen Zahlenwert
 Lesen = 4, Schreiben = 2, Ausführen = 1

Sie haben drei 3er Gruppen mit Attributen




Das erste Triple sagt aus, was der Eigentümer einer Datei damit machen darf

Das zweite Triple beschreibt, was die Eigentümer-Gruppe einer Datei damit machen darf

Das dritte Triple beschreibt, was die „Welt“, also alle anderen außer Eigentümer und Gruppe damit machen dürfen.

Die Kombination der Werte mit Eigentümer und Gruppe ergibt die Berechtigung.

Zur Vertiefung: <https://linuxwiki.org/DateiRechte>

Linux – Datei- und Benutzerrechte (2)

Rechte: read, write, execute / change to; Numerik: r = 4, w = 2, x = 1

owner	group	other	ergibt
r	-	-	400
rw	-	-	600
rwX	-	-	700
rwX	r	-	740
rwX	r-X	-	750
rwX	rwX	-	770
rwX	rwX	r	774
rwX	rwX	r-X	775
rwX	rwX	rwX	777
rwX	r-X	r-X	755

chmod [MODUS] [Datei], z.B. chmod 755 datei.txt

14.11.23
Horst Bräuner
17

Die Rechte werden durch ihre Zahlenwerte repräsentiert. Die einzelnen Werte werden addiert. Hier nochmal ein Beispiel:

Rechte: r = 4, w = 2, x = 1
 Datei: datei.txt
 Eigentümer Hugo
 Gruppe MIB
 Mitglieder der Gruppe MIB = Helga, Max

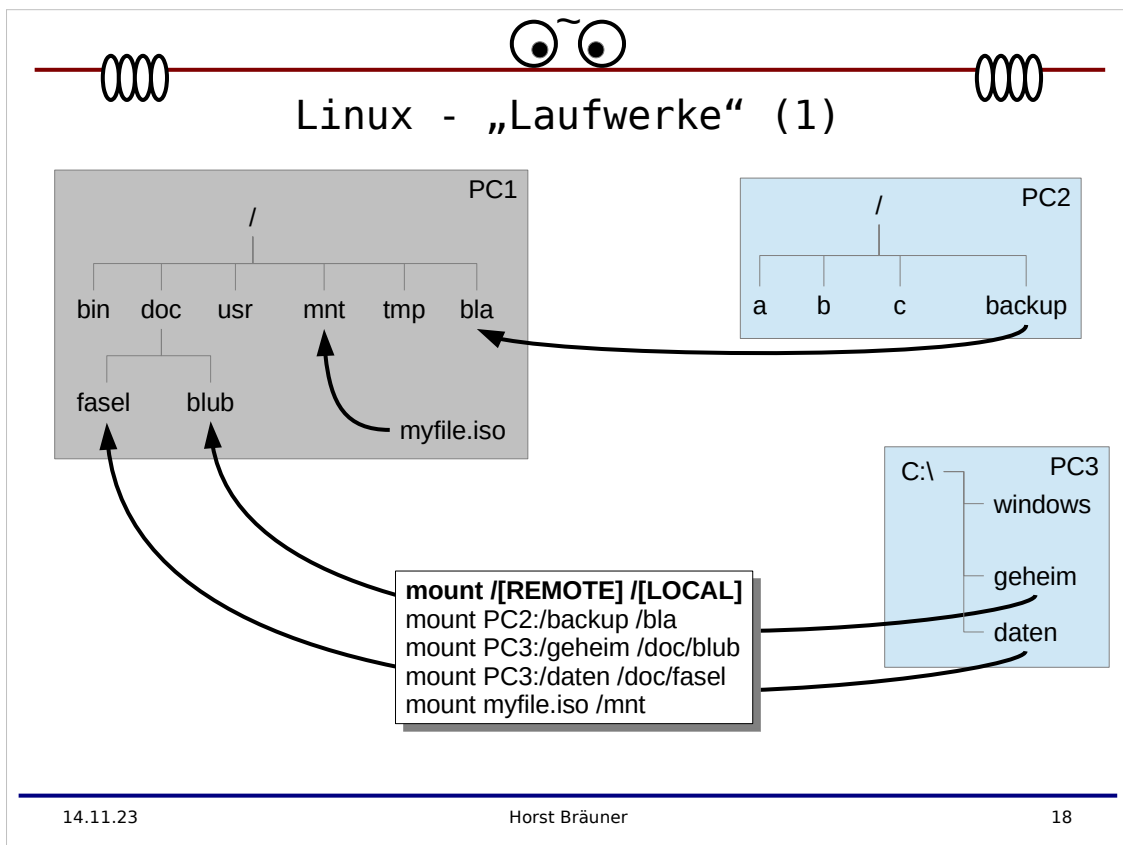
rwX - - datei.txt bedeutet dann: Hugo darf Lesen, Schreiben, Ausführen und das Dateirecht wurde gesetzt mich „chmod 700 datei.txt“

rwX r - datei.txt bedeutet dann: Hugo darf Lesen, Schreiben, Ausführen; Max und Helga dürfen die Datei Lesen und das Dateirecht wurde gesetzt mit „chmod 740 datei.txt“

Achtung:

rwX - r datei.txt bedeutet dann: Hugo darf Lesen, Schreiben, Ausführen; Max und Helga dürfen die Datei ebenfalls Lesen, da sie über „Rest der Welt“ das Leserecht erhalten. Rechte sind NICHT ausschließend. Das Dateirecht wurde gesetzt mit „chmod 704 datei.txt“

Selbstverständlich sind nicht alle Kombinationen von Rechten sinnvoll ;-)

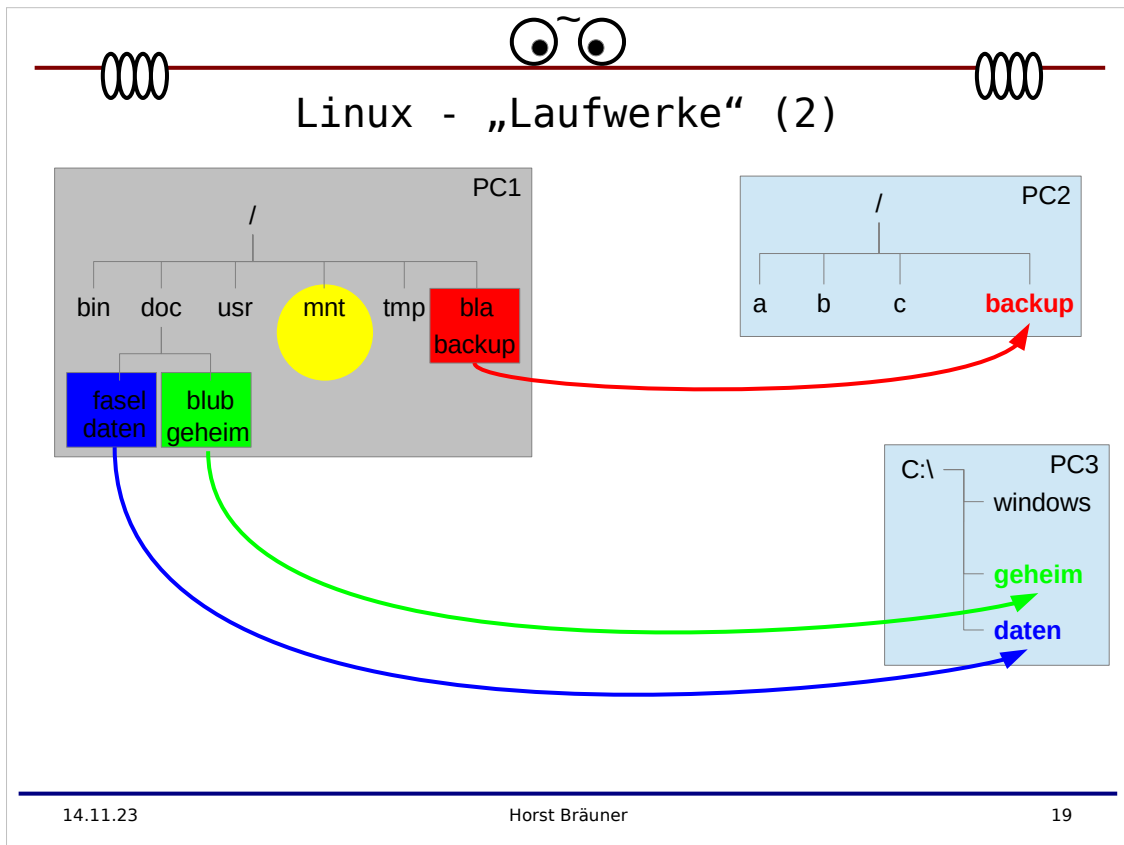


Laufwerke, Netzlaufwerke

Linux kennt keine Laufwerksbuchstaben.


Freigaben anderer Rechner können an beliebiger Stelle „eingehängt“ (ge-“mounted“) werden und sehen dann aus wie lokale Laufwerke.

Im obigen Beispiel „sieht“ der Anwender auf PC1 die Verzeichnisse fasel, blub und bla jedoch sieht er auf den ersten Blick nicht, dass sie auf fremden Rechnern liegen.

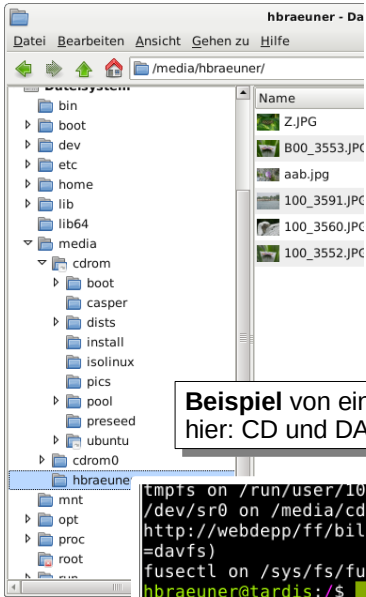


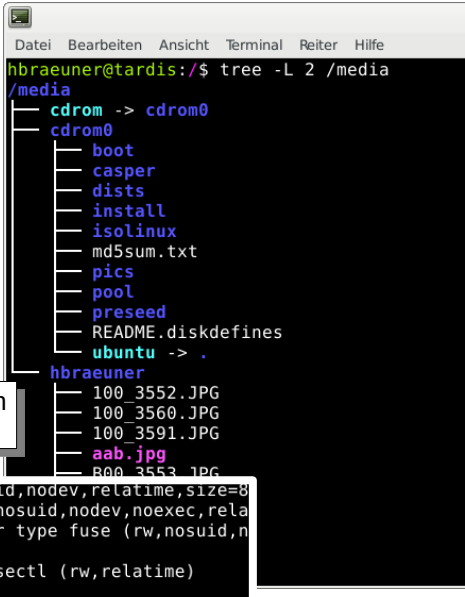
Hier noch einmal optisch dargestellt. Der Anwender auf PC1 „sieht“ die Verzeichnisse auf den fremden Rechnern nicht mit den dortigen Namen, sondern mit den Namen an deren Stelle sie in seinem eigenen Rechner eingehängt wurden.

Das Einhängen von Verzeichnissen ist grundsätzlich Aufgabe eines Administrators. „Normale“ Benutzer haben nicht das Recht dazu.



Linux - „Laufwerke“ (3)





14.11.23
Horst Bräuner
20

Noch ein Beispiel aus der Praxis:

Sie sehen links in dem Dateibaum das Verzeichnis „media“, darunter das Verzeichnis „cdrom“ bzw. „cdrom0“. An diese Stelle wurde das Gerät /dev/sr0 (= CD-Laufwerk) eingehängt.

Weiter unten im Dateibaum sehen Sie „hbraeuner“. Das ist ein Laufwerk, das von einem anderen Rechner via „http“ verbunden wurde.

Im unteren Ausschnitt aus der Kommandozeile sehen Sie beide „mounts“ nochmal dev/sr0 an media/cdrom0 und http://webdepp/ff/bilder an /media/hbraeuner.



Linux - Konfigurationsdateien

```
<VirtualHost *:80>
    ServerName kasperl

    ServerAdmin webmaster@kasperl
    DocumentRoot /var/www/kasperl/
    <Directory "/var/www/kasperl/">
        DAV on
        Options +Indexes
    <Limit PROPFIND>
        #wegen BUG 54914
        DirectoryIndex foo.bar
    </Limit>
    </Directory>

    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/kasperlerror.log
    CustomLog ${APACHE_LOG_DIR}/kasperlaccess.log
</VirtualHost>
root@webdepp:~#
```

Konfiguration eines Webserver
(in Auszügen)

```
# This file describes the network interfaces
# and how to activate them. For more informat
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp




auto eth1
iface eth1 inet static
    address 192.168.56.66
    netmask 255.255.255.0
root@webdepp:~#
```

Konfiguration einer Netzwerkkarte
eth0 mit DHCP
eth1 manuell

Ein Beispiel für Konfigurationsdateien. Der Inhalt ist hier nicht wichtig. Das Prinzip sollte verstanden werden.

Links ein Beispiel für die Konfiguration eines Webserver. Diese wird Ihnen beim Thema Webserver detailliert erläutert.

Rechts ein Beispiel, wie das Netzwerk auf einem Linux-System konfiguriert werden kann. Es gibt selbstverständlich (oder leider?) noch andere Methoden. eth0 und eth1 sind die beiden Netzwerkkarten des Rechners.



Linux – Befehlszeile und Scripte (1)

```
user@host:~$ ls
DataSpace.XXX      labyrinth2.odp      SAM_1253.JPG
DaTei              LfdDBW.odt         SAM_1254.JPG
*datei             LfVBW.odt          Schreibtisch
datei6             lkj                 sha
datei7             llauf.odt          shs2ip51-config
daTei.txt          mail               ssl
DATEi.txt          Maildir            test
dnsplan.png        meine              test.sh
Dokumente          meinedaten         ticket-20150930073339.pdf

user@host:~$ cat DATEi.txt
Hallo Hallo
Hallo Hallo
Hallo Hallo

user@host:~$ /sbin/ifconfig -a | grep -i up
UP BROADCAST MULTICAST MTU:1500 Metrik:1
UP BROADCAST RUNNING MULTICAST MTU:1500 Metrik:1
Interup:11 Basisadresse:0xc000 Speicher:c0210000-c0210fff
UP LOOPBACK RUNNING MTU:16436 Metrik:1
```

user@host:~\$

Horst Bräuner

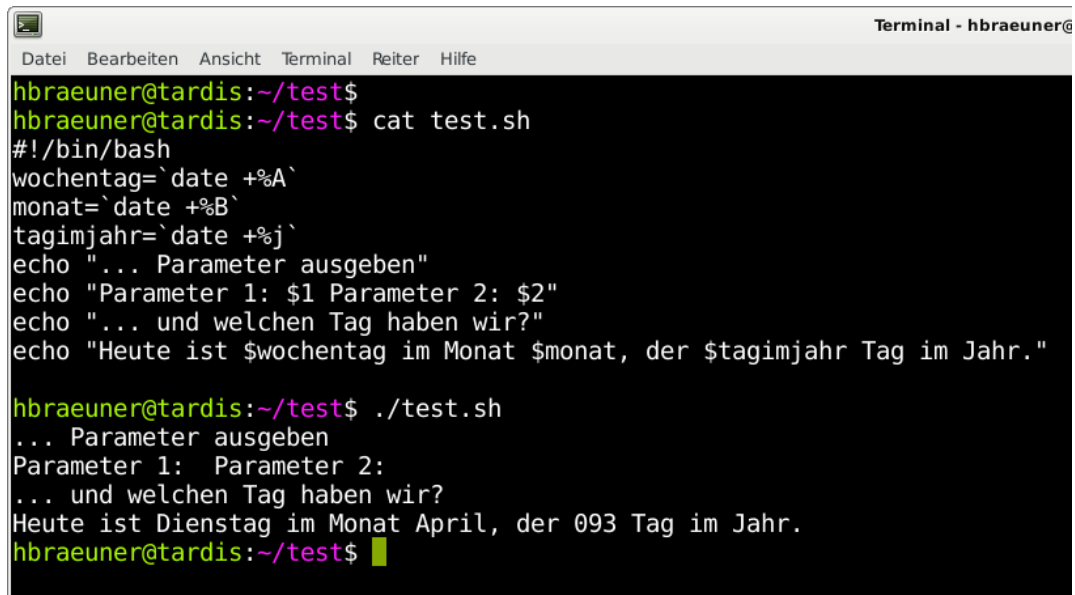
22

An der Befehlszeile können Sie, wie der Name schon sagt, Kommandos absetzen. Ein Liste mit oft verwendeten Befehlen erhalten Sie separat.

Jedes Kommando / jeder Befehl kann mit „Schaltern“ modifiziert werden. Zum Beispiel bedeutet „ls“ - zeige mir die Dateien im aktuellen Verzeichnis, „ls -la“ - zeige mir die Dateien im aktuellen Verzeichnis ABER ALLE, auch die verborgenen.

Im unteren Beispiel werden Kommandos / Befehle verknüpft. „ifconfig“ zeigt die Netzwerkkarten, „grep“ sucht nach Mustern. Die Kombination mit dem „|“ bewirkt, dass innerhalb des Kommandos „Zeige Netzwerkkarten“ nach dem Wert „up“ gesucht wird. Das Ergebnis ist eine Liste mit Netzwerkkarten die aktiv sind.

Linux – Befehlszeile und Scripte (2)

A terminal window titled "Terminal - hbraeuner@" with a menu bar (Datei, Bearbeiten, Ansicht, Terminal, Reiter, Hilfe). The prompt is hbraeuner@tardis:~/test\$. The user runs 'cat test.sh', displaying a script that sets variables for day of week, month, and day of year, and prints them with some explanatory text. Then the user runs './test.sh', and the script's output is shown, indicating it is Tuesday in April, the 093rd day of the year.

```
hbraeuner@tardis:~/test$  
hbraeuner@tardis:~/test$ cat test.sh  
#!/bin/bash  
wochentag=`date +%A`  
monat=`date +%B`  
tagimjahr=`date +%j`  
echo "... Parameter ausgeben"  
echo "Parameter 1: $1 Parameter 2: $2"  
echo "... und welchen Tag haben wir?"  
echo "Heute ist $wochentag im Monat $monat, der $tagimjahr Tag im Jahr."  
  
hbraeuner@tardis:~/test$ ./test.sh  
... Parameter ausgeben  
Parameter 1: Parameter 2:  
... und welchen Tag haben wir?  
Heute ist Dienstag im Monat April, der 093 Tag im Jahr.  
hbraeuner@tardis:~/test$
```

14.11.23

Horst Bräuner

23


Ein Beispiel für ein script. Dieses gibt Informationen zu übergebenen Parametern und zum aktuellen Tag aus.

Sie sehen daran, dass Scripte ähnlich wie eine klassische Programmiersprache arbeiten. Falls Sie sich näher damit beschäftigen möchten, empfehle ich

https://wiki.ubuntuusers.de/Shell/Bash-Skripting-Guide_für_Anfänger/

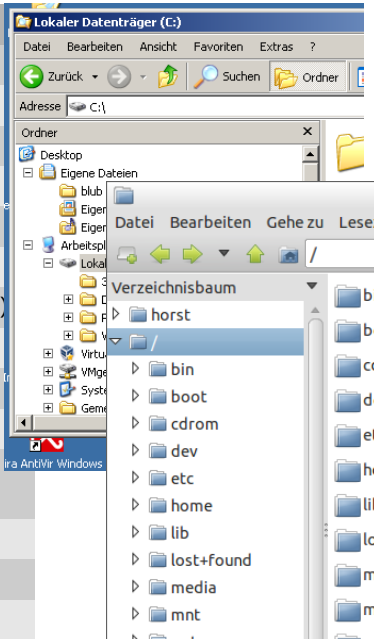
oder

<http://www.linux-praxis.de/lpic1/lpi102/1.109.2.html>



Linux - Verzeichnisbaum

Verzeichnis	Inhalt
bin	Programme („binaries“) für „normale“ Benutzer
boot	Bootmanager und Kernel
dev	Geräte Dateien („Beschreibungen“)
etc	Konfigurationsdateien
home	Benutzerverzeichnisse
lib	Bibliotheken für dynamisch gelinkte Programme
lost+found	für „verlorene“ Dateien nach Filesystem-checks (ext2/3/4)
mnt	leeres Verzeichnis als Standard-„mountpoint“
opt	optionale Software, z.B. Acrobat Reader, Browser, ...
proc	Pseudo-Verzeichnis für laufende Prozesse
sbin	Systemprogramme, die eigentlich nur ein privilegierter Benutzer („root“) ausführen darf
tmp	temporäre Dateien
usr	Programme, die nicht unbedingt „lebensnotwendig“ sind
var	variabler Inhalt, z.B. „spool“-Dateien und „log“s



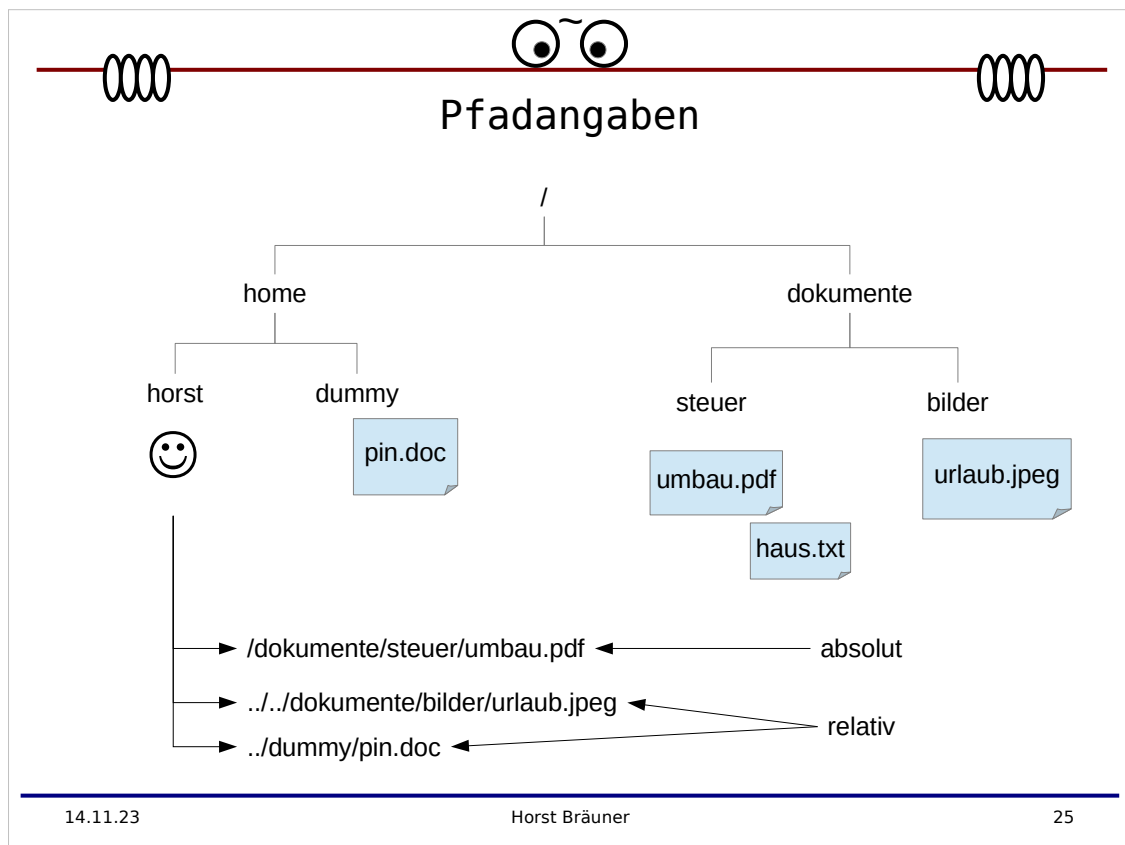
14.11.23
Horst Bräuner
24

Der grundsätzliche Aufbau des Linux-Verzeichnisbaums

Theoretisch sollte dieser selbsterklärend und verbindlich sein, aber in der Praxis hält sich nicht jeder Entwickler an dieses Schema.

Wichtig zu wissen

- Geräte liegen unterhalb von „dev“, z.B. Festplatten /dev/disk/...
- Nahezu alle Konfigurationsdateien liegen unterhalb von „etc“.
- Benutzerverzeichnissen liegen unterhalb von „home“.



Wie auch in anderen Betriebssystemen können Pfade zu Dateien relativ und absolut angegeben werden. Im Beispiel befinden Sie sich an der Stelle des Smiley.

Noch ein Beispiel:

Absoluter Pfad

/home/datei1.txt

/home/hugo/datei2.txt

/home/hugo/MeineDateien/datei3.txt

Wenn Sie im Verzeichnis /home/hugo/MeineDateien/ sind, sind die relativen Pfade zu den Dateien ...

./datei3.txt = die Datei ist im aktuellen Verzeichnis

../datei2.txt = die Datei ist ein Verzeichnis über dem aktuellen Verzeichnis

../../datei1.txt = die Datei ist zwei Verzeichnisse über dem aktuellen Verzeichnis



Linux – Systemverwaltung

- Systemverwalter Account
- wird bei der Installation automatisch angelegt
- besitzt **alle** Rechte im System
- kann weitere Benutzer einrichten und verwalten

Ich bin root – Ich darf das

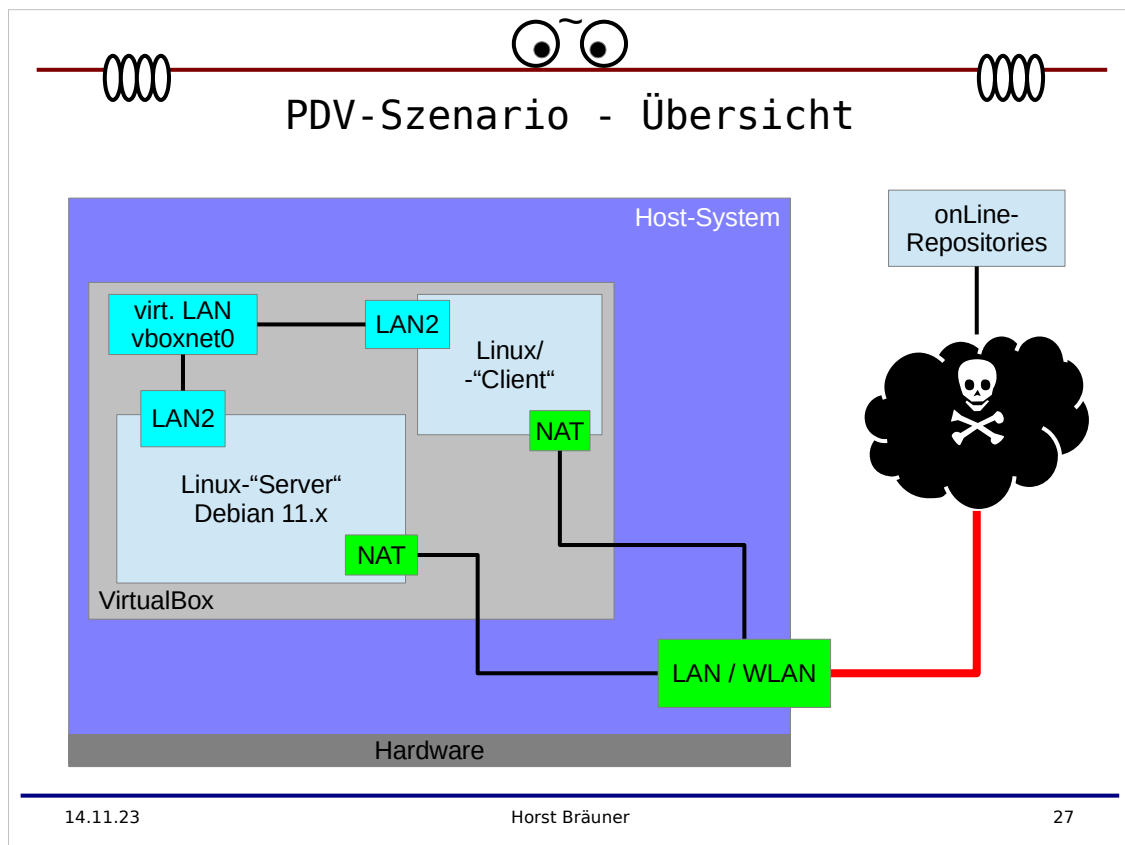
- arbeitet immer mit Privilegien !!!
- mit Vorsicht zu verwenden !!!
- wird unterstellt, dass sie/er weiß, was sie/er tut !!!
- praktisch keine Rückfragen !!!

In allen Unix/Linux Betriebssystemen gibt es grundsätzlich nur einen Administrator den „root“.

Dieser Benutzer hat alle Rechte und darf alles tun. Ihm wird unterstellt, dass er weiß, was er tut.

Es gibt praktisch keine Rückfragen „sind Sie sicher?“, „sind sie wirklich sicher?“, „sind Sie absolut sicher?“. Wenn root eine Datei löscht, ist sie grundsätzlich weg. Es gibt keinen Papierkorb oder eine „Rückgängig“ Funktion. Wenn Sie als root arbeiten und ein System löschen, ist das System tatsächlich zerstört.

Deshalb seien Sie in der Praxis vorsichtig, wenn Sie als root arbeiten.



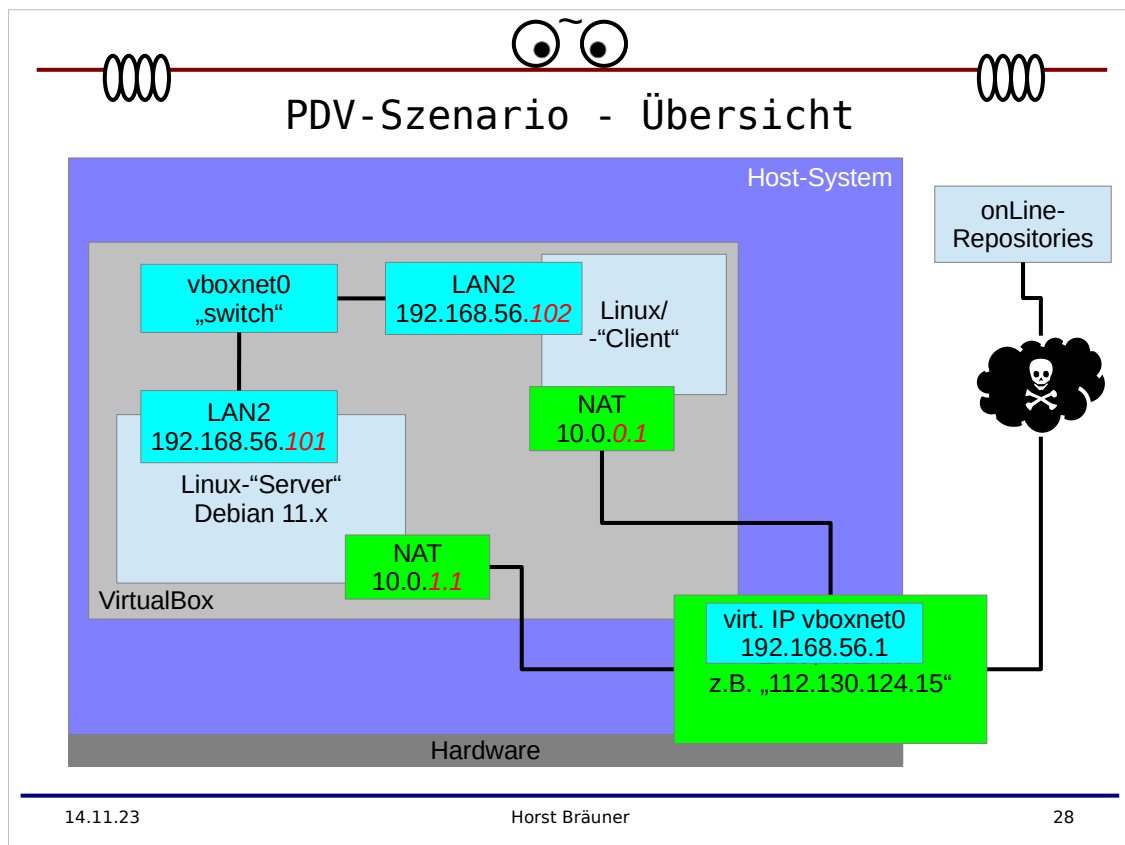
Das Szenario, wie wir es in dieser Vorlesung verwenden.

Wir installieren alle Systeme als virtuelle Maschinen (VM). „Host“-System ist Ihr Laptop oder Rechner.

Ich verwende als Virtualisierungslayer VirtualBox (www.virtualbox.org). Sie können auch HyperV oder VmWare benutzen, falls Sie mögen. Dann müssen Sie die Anbindung an Internet und Ihr Host-System entsprechend Ihrer Virtualisierung vornehmen.

Wir installieren 1 VM mit Debian 12 (nächstes Kapitel), die wir dann zum klonen für unsere weiteren virtuellen Linuxe nutzen. Unser erstes virtuelles Linux wird mit einer „NAT“-Netzwerkkarte installiert, die transparent durch die Netzwerkkarte Ihres Host-Systems auf das Internet zugreifen kann.


Später fügen wir eine Netzwerkkarte hinzu, die unser virtuelles Linux mit einem weiteren virtuellen Linux und/oder Ihrem Host-System verbindet.




Unter VirtualBox sieht die Konfiguration mit IP-Adressen am Ende so aus. Die Konfiguration wird später in einzelnen Schritten erklärt.

Jedes virtuelle Linux belegt maximal 8GByte Festplatte auf Ihrem System.

Falls Sie „Platzprobleme“ :-)) auf Ihrem Host-System haben, kommen Sie auch mit nur einer virtuellen Linux-Maschine aus. Die Netzwerkverbindungen realisieren Sie dann mit Ihrem Host-System oder über „localhost“ (siehe „IP“ vorne im Script).

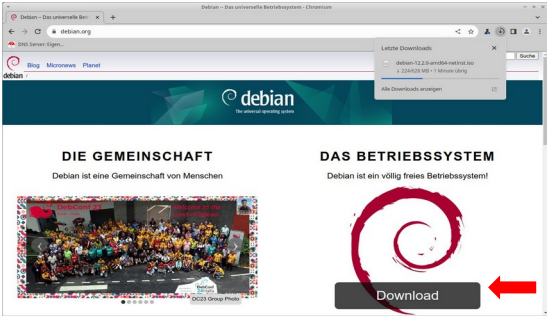


PDV-Szenario - Software



www.virtualbox.org
 - VirtualBox herunterladen und installieren
 - **extension-pack** installieren, wegen USB
 - aktueller User ggf. zu Gruppe „vboxusers“ hinzufügen

www.debian.org
Netinstall .iso herunterladen



14.11.23
Horst Bräuner
29

Holen und installieren Sie sich VirtualBox und holen Sie eine Installations-CD von Debian. Sofern Sie mit aktiver Internet-Verbindung installieren, genügt die Minimale „netinstall“ CD.

VirtualBox können Sie unter

<https://www.virtualbox.org/wiki/Downloads>

für Ihr System passend herunterladen. Installieren Sie bitte auch das VirtualBox Extension Pack.

Die Debian „.iso“ Datei laden Sie von

<https://www.debian.org/distrib/netinst>

für Ihr System passend herunter. „amd64“ falls Sie ein 64-Bit Host-System haben und „i386“ für ein 32-Bit Host-System.