



Feedback aus der Hausaufgabe

- Was ist Ihnen aufgefallen?
- Gab es grundlegende neue Erkenntnisse?
- Was hat gefehlt?
- Wieviel Zeit haben Sie aufgewendet?

Lektion 9: Virtualisierungstechnologien und Systemkern-Konfiguration / Tuning



Vorbereitung für die heutige Übung

- Starten Sie möglichst rasch den Download der Datei «Linux-Mint_18.3_Cinnamon-VB-32bit.7z» aus dem Unter-Ordner «Übungen» auf dem Studierenden AD-Server – die Datei ist sehr gross (1.2 Gbyte) und sollte für die Übung auf ihrem Rechner bereitstehen.
- Entpacken Sie die Datei mit dem Gratis-Programm «7Zip» (Windows Download unter «<https://www.7zip.org/>», Linux: «sudo apt-get install p7zip-full», dann «man 7z» für die Bedienungsanleitung; Apple IOS: «unarchiver» installieren/benutzen)
- Das Resultat ist eine ca. 6 Gbyte gross «.vdi-Datei»
→ bitte lokal abspeichern für die Übung

Grundlagen der Virtualisierung



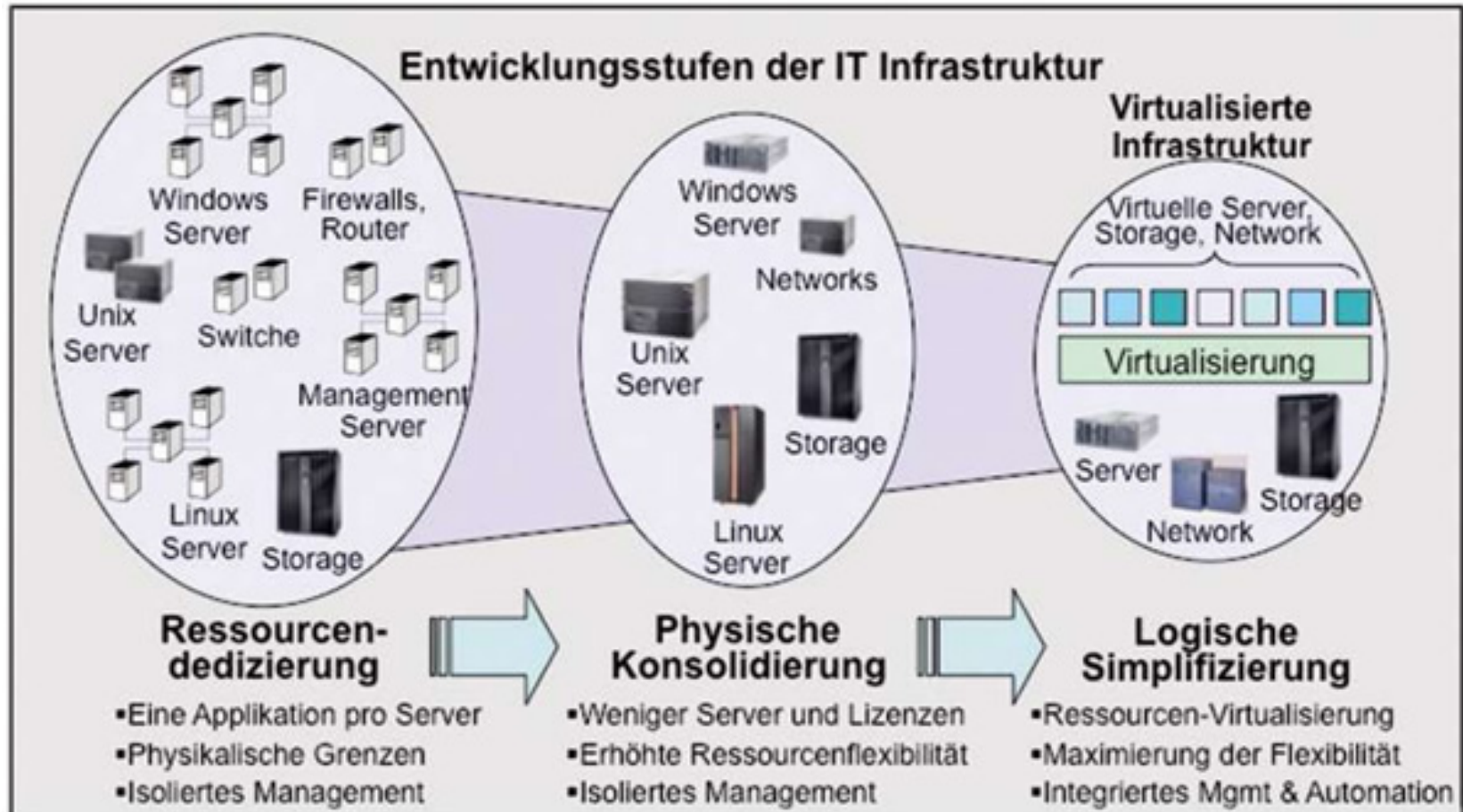
© Scott Adams, Inc./Dist. by UFS, Inc.



Grundlagen der Virtualisierung

- Die Einschränkungen der heutigen Server, die auf die Ausführung jeweils nur eines Betriebssystems und einer Anwendung ausgelegt sind, stellen IT-Abteilungen vor grosse Probleme. So müssen selbst kleine Rechenzentren viele Server bereitstellen, die jeweils nur zwischen 5 und 15 Prozent ausgelastet und damit hochgradig ineffizient sind.
-
- Das Prinzip der Virtualisierung ist einfach: Mithilfe von Software wird das Vorhandensein von Hardware simuliert und ein virtuelles Computersystem erstellt. Auf diese Weise können Unternehmen mehrere virtuelle Systeme – und mehrere Betriebssysteme und Anwendungen – auf einem einzigen Server ausführen. So können Grössenvorteile und eine höhere Effizienz erzielt werden.

<https://www.vmware.com/ch/solutions/virtualization.html>



http://i.cmpnet.com/informationweek.de/iwk_img/2009/03/iwk_03_10_01.jpg

- Ein virtuelles Computersystem – die so genannte virtuelle Maschine (VM) – ist ein vollständig isolierter Software-Container mit einem Betriebssystem und einer Anwendung. Jede eigenständige VM ist völlig unabhängig. Die Nutzung mehrerer VMs auf einem einzigen Computer ermöglicht die Ausführung mehrerer Betriebssysteme und Anwendungen auf nur einem physischen Server oder „Host“.
- Mittels einer schlanken Softwareschicht – dem so genannten Hypervisor – werden die virtuellen Maschinen vom Host abgekoppelt. Jeder einzelnen virtuellen Maschine werden bei Bedarf dynamisch Computing-Ressourcen zugeteilt.

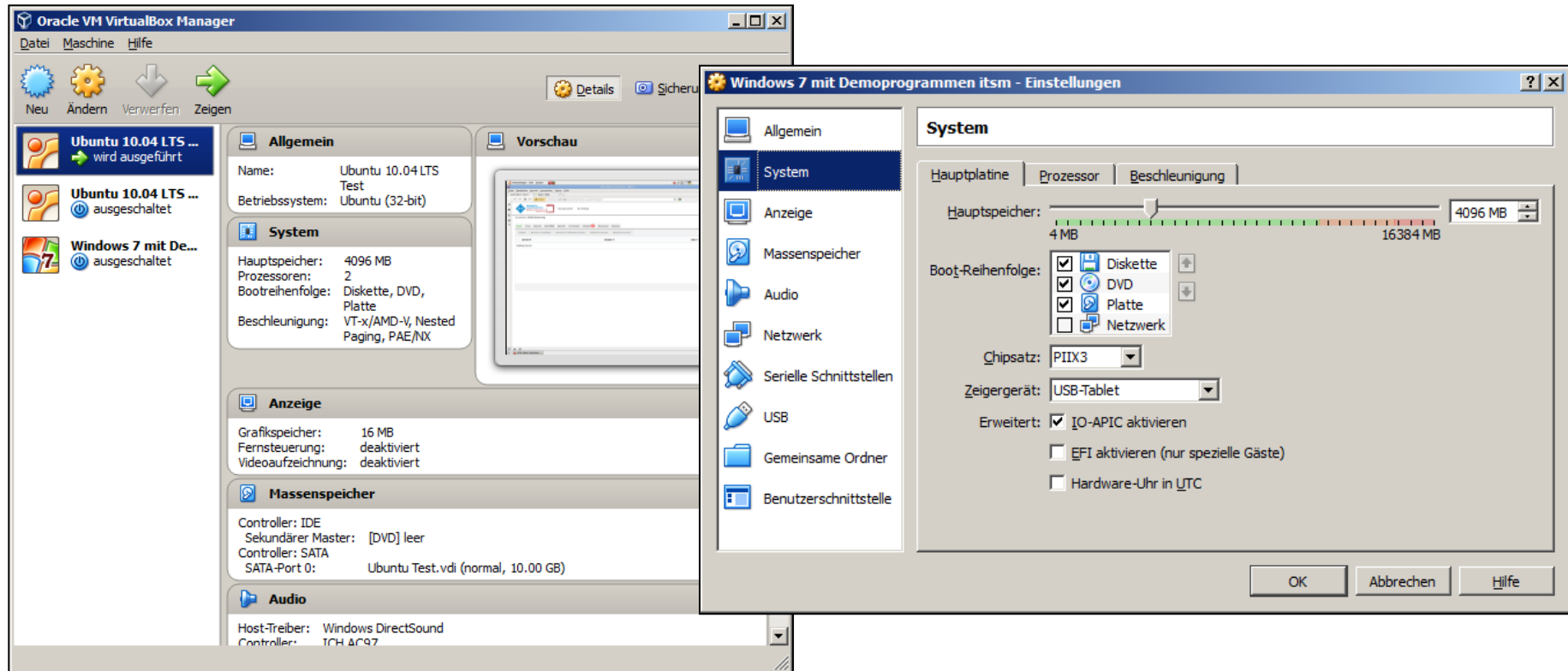
<https://www.vmware.com/ch/solutions/virtualization.html>

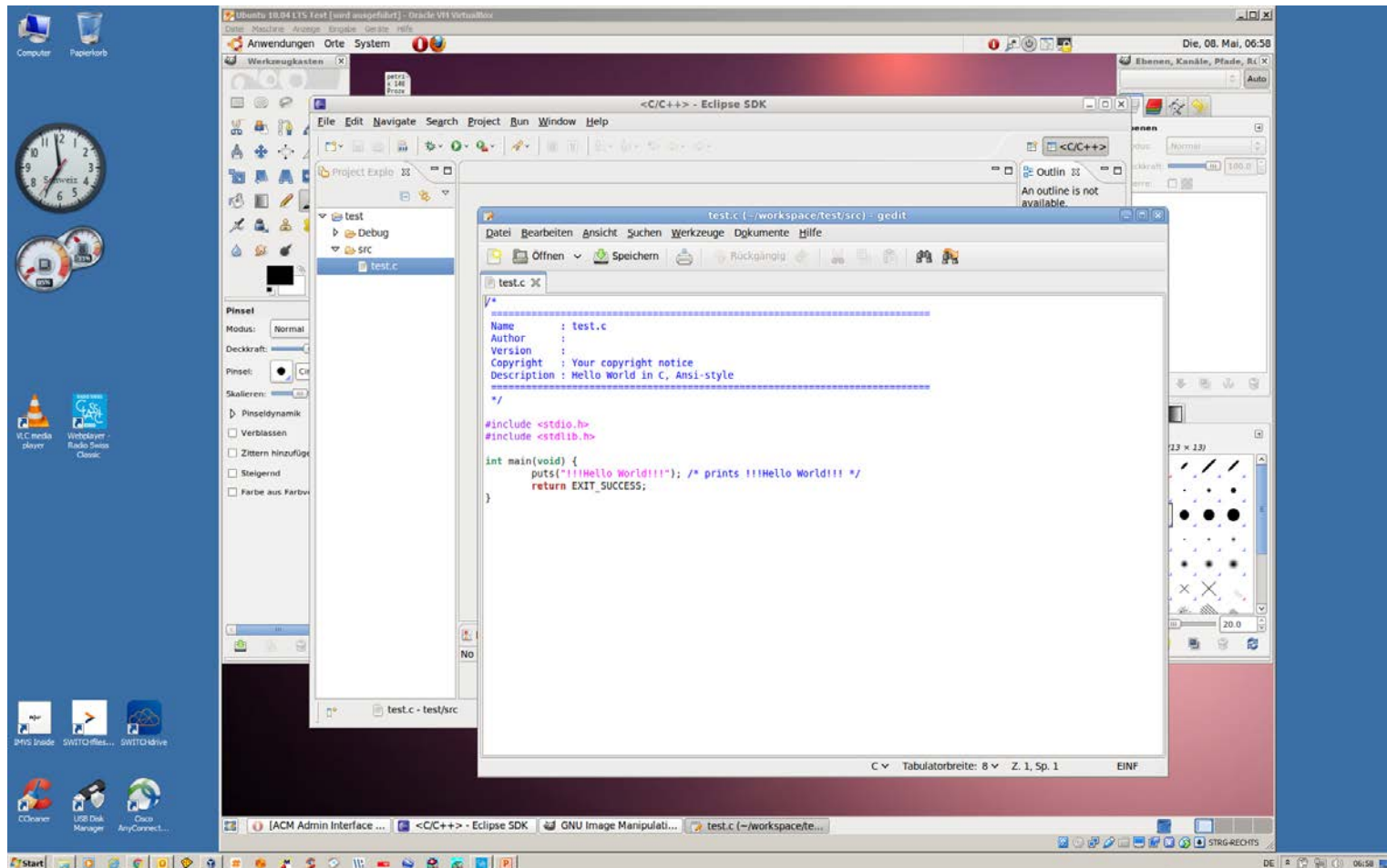
Eigenschaften virtueller Maschinen

- **Partitionierung**
 - Ausführen mehrerer Betriebssysteme auf einem einzigen physischen Computer
 - Aufteilen von Systemressourcen zwischen virtuellen Maschinen
- **Isolation**
 - Fehler- und Sicherheitsisolation auf Hardwareebene
 - Erweiterte Ressourcensteuerung für gleichbleibende Performance
- **Kapselung**
 - Speichern des gesamten VM-Zustands in Dateien
 - Unkompliziertes Verschieben und Kopieren von virtuellen Maschinen (so einfach wie von Dateien)
- **Hardwareunabhängigkeit**
 - Bereitstellung oder Migration jeder virtuellen Maschine auf jedem bzw. jeden beliebigen physischen Server

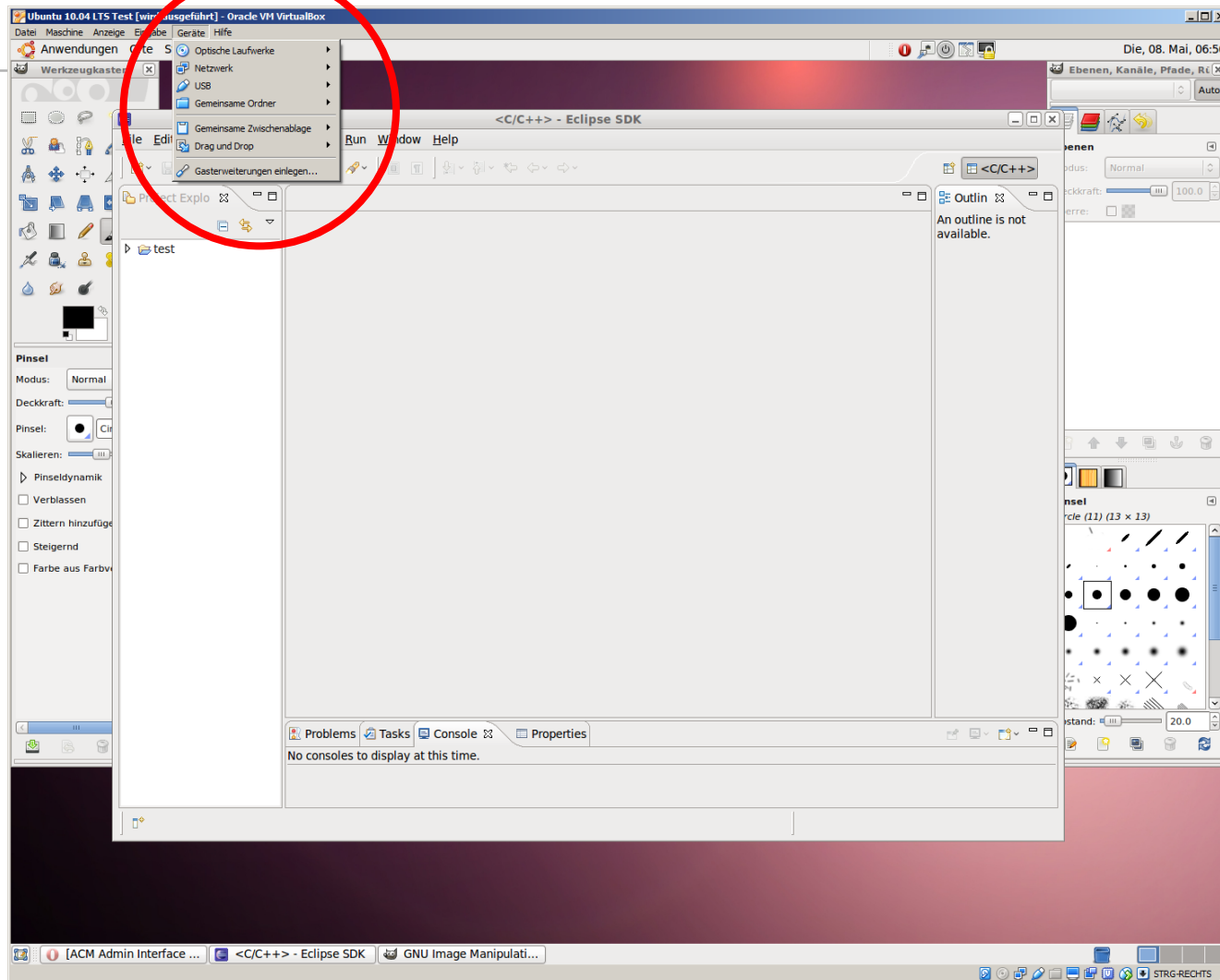
<https://www.vmware.com/ch/solutions/virtualization.html>

- Im Besitz von Oracle Inc., derzeit gratis





Beispiel: VirtualBox



Vorteile der Virtualisierung

- Virtualisierung erhöht die Agilität, Flexibilität und Skalierbarkeit der IT und ermöglicht dabei deutliche Kosteneinsparungen. IT-Komponenten lassen sich einfacher verwalten und kostengünstiger betreiben, da Workloads schneller bereitgestellt, Performance und Verfügbarkeit optimiert und Betriebsabläufe automatisiert werden.
- Weitere Vorteile:
 - Reduzierte Investitions- und Betriebskosten
 - Minimale oder keine Ausfallzeiten
 - Erhöhte Produktivität, Effizienz, Agilität und Reaktionsfähigkeit der IT
 - Schnellere und einfachere Anwendungsbereitstellung
 - Business Continuity und Disaster Recovery
 - Einfacheres Management von Rechenzentren
 - Aufbau eines echten Software-Defined Datacenter

<https://www.vmware.com/ch/solutions/virtualization.html>

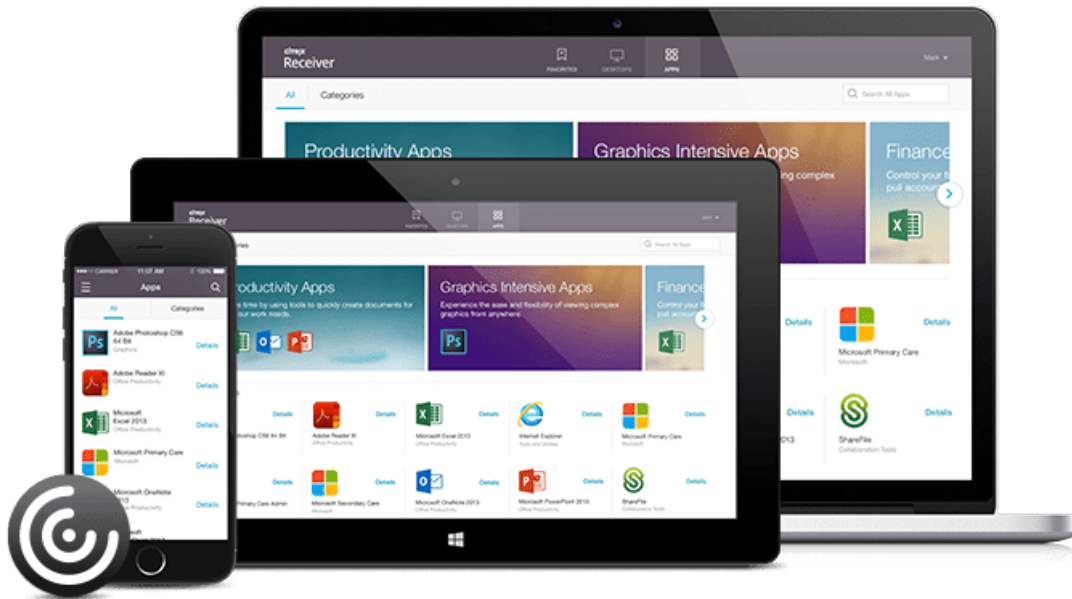
Risiken der Virtualisierung

- Erhöhte Komplexität der Gesamtlösung
- Kritische Pfade / Abhängigkeitsketten sind schwerer zu erkennen
- Zusatzmechanismen für die Ausfallsicherheit
- Erschwerte Überwachung und Fehlersuche / -behebung
- Gegenseitige Einflüsse zwischen VMs aufgrund fehlerhafter Konfiguration oder Schwachstellen der Software

- **Serverkonsolidierung**
 - Durch Server-Virtualisierung kann ein Unternehmen die Auslastung der Serverressourcen maximieren und die Anzahl benötigter Server reduzieren. Das Ergebnis ist eine Serverkonsolidierung, die zu einer verbesserten Effizienz und niedrigeren Kosten beiträgt.
- **Kein Cloud Computing**
 - Cloud Computing und Virtualisierung sind nicht dasselbe. Vielmehr bildet Virtualisierung die Basis für Cloud Computing. Cloud Computing bezeichnet die Bereitstellung gemeinsam genutzter Computing-Ressourcen (Software und/oder Daten) nach Bedarf über das Internet. Unabhängig davon, ob man eine Cloud nutzt oder nicht, kann man Server virtualisieren und sich später für Cloud Computing entscheiden, wenn man die Agilität und Self-Service-Funktionen weiter verbessern möchte.

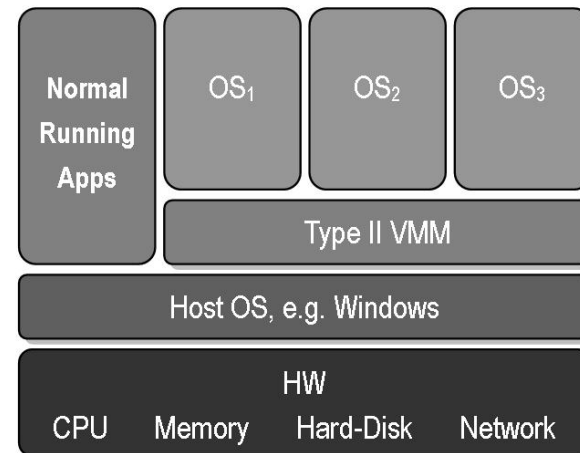
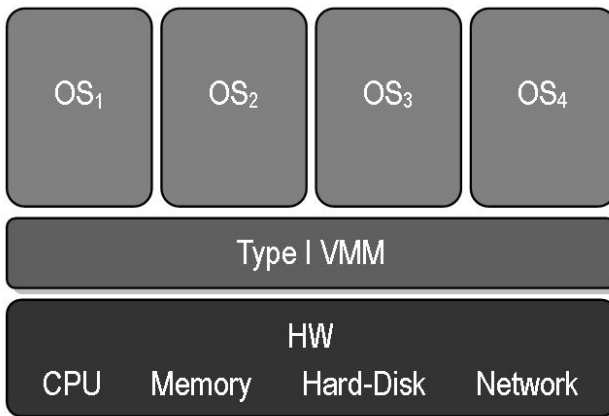
<https://www.vmware.com/ch/solutions/virtualization.html>

- Es können nicht nur Server, sondern auch Clients virtualisiert werden.
- Im nächsten Schritt werden einzelne Anwendungen auf dem Desktop virtualisiert.
- Anwendungsbeispiel an der FHNW: vdesk.fhnw.ch (Citrix Receiver)



Der Hypervisor

Hypervisoren erlauben den simultanen Betrieb mehrerer Gastsysteme auf einem Hostsystem. Der Hypervisor verwaltet die Ressourcenzuteilung für einzelne Gastsysteme. Er verteilt die Hardware-Ressourcen derart, dass für jedes einzelne Gastbetriebssystem alle Ressourcen bei Bedarf verfügbar sind, so, als ob nur ein Betriebssystem vorhanden wäre ... Den einzelnen Gastsystemen wird dabei jeweils ein eigener kompletter Rechner mit allen Hardware-Elementen (Prozessor, Laufwerke, Arbeitsspeicher usw.) vorgespielt. (Wikipedia)



Applikations- Virtualisierung: Docker/Container

- Docker ist eine Open-Source-Software zur Isolierung von Anwendungen mit Container-Virtualisierung.
- Docker vereinfacht die Bereitstellung von Anwendungen, weil sich Container, die alle nötigen Pakete enthalten, leicht als Dateien transportieren und installieren lassen. Container gewährleisten die Trennung und Verwaltung der auf einem Rechner genutzten Ressourcen. Das beinhaltet laut Aussage der Entwickler: Code, Laufzeitmodul, Systemwerkzeuge, Systembibliotheken – alles was auf einem Rechner installiert werden kann.
(Wikipedia)

- Der Docker-Container, und damit auch die in ihm enthaltenen Anwendungen, wird architekturbedingt als *root* oder in einer eigenen Nutzergruppe ausgeführt, die *root* gleichgestellt ist und damit uneingeschränkten Zugang zu allen Betriebssystemfunktionen hat.
- Gelingt es einem Angreifer, in der Virtualisierung *Superuser*-Rechte zu erlangen, so kann er diese nicht im Host geltend machen, da dieser aufgrund der Linux-Kernel-Funktion der Namensräume nicht vom Container aus erreichbar ist. Dieses Verfahren wird bereits seit 2008 in den Docker-ähnlichen Linux-LXC-Containern erfolgreich verwendet.
- Entgegen dem Konzept von Virtualisierung werden die Host-Ressourcen nicht durch Virtualisierung der Hardware, sondern durch das Rechtemanagement geschützt (Sandbox). (Wikipedia)

- rkt - Rocket (CoreOS)
- LXC
- Linux Vserver
- OpenVZ/Virtuozzo 7
- runC
- FreeBSD Jail
- Oracle Solaris Zones
- Windows Server 2018 Docker Port
- Microsoft Drawbridge
- Windocks
- Sandboxie
- Turbo / Spoon
- VMWare ThinApp

<https://hosting.1und1.de/digitalguide/server/knowhow/docker-alternativen-im-ueberblick/>

Zusammenfassung Virtualisierung

- Spannende Technologie und sinnvolle Einsatzgebiete (u.a. Vorbedingung für Cloud-basierte Dienste)
aber:
- Egal, wie viel ich virtualisiere – irgendwo steht «das Blech» und muss betrieben, überwacht, gewartet werden
- Der Ressourcen-Pool bleibt endlich
- Auswirkungen auf Betriebssysteme – CPU, Speicher, Ein-/Ausgabe, Treiber usw., auch durch den Hypervisor
- Das gehostete System weiss (kann herausfinden), dass es gehostet wird – unterschiedliches Verhalten
- Komplexität, Abhängigkeitsketten, Sicherheitsfragen



Übung (ca. 30 min.)

- Aufgabe(n) gemäss separatem Aufgabenblatt
- Lösungsansatz: Einzelarbeit oder Gruppen von max. 3 Personen
- Hilfsmittel: beliebig
- Besprechung möglicher Lösungen in der Klasse (es gibt meist nicht die eine «Musterlösung»)

Übungsbesprechung (ca. 15 min.)

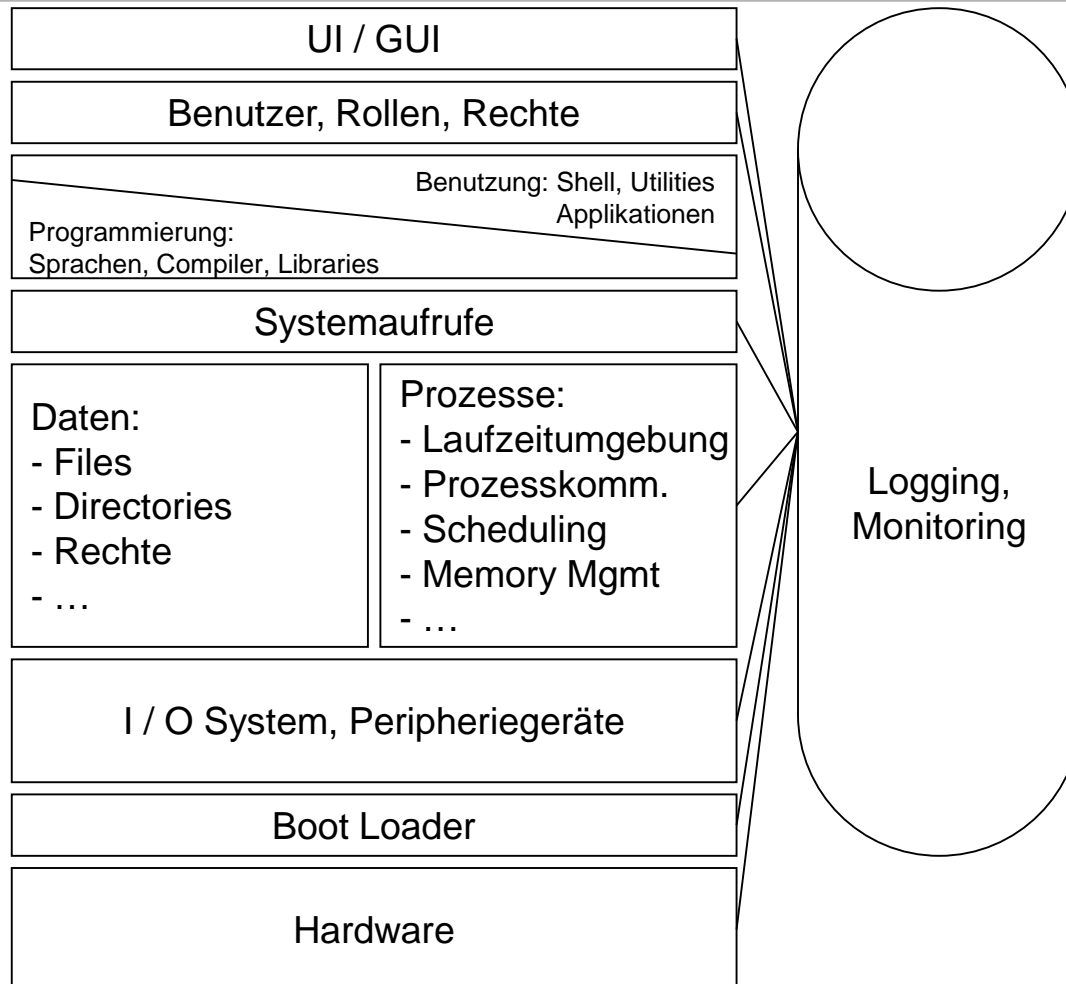
- Stellen Sie Ihre jeweilige Lösung der Klasse vor.
- Zeigen Sie auf, warum ihre Lösung korrekt, vollständig und effizient ist.
- Diskutieren Sie ggf. Design-Entscheide, Alternativen oder abweichende Lösungsansätze.
- Gibt es Unklarheiten? Stellen Sie Fragen.



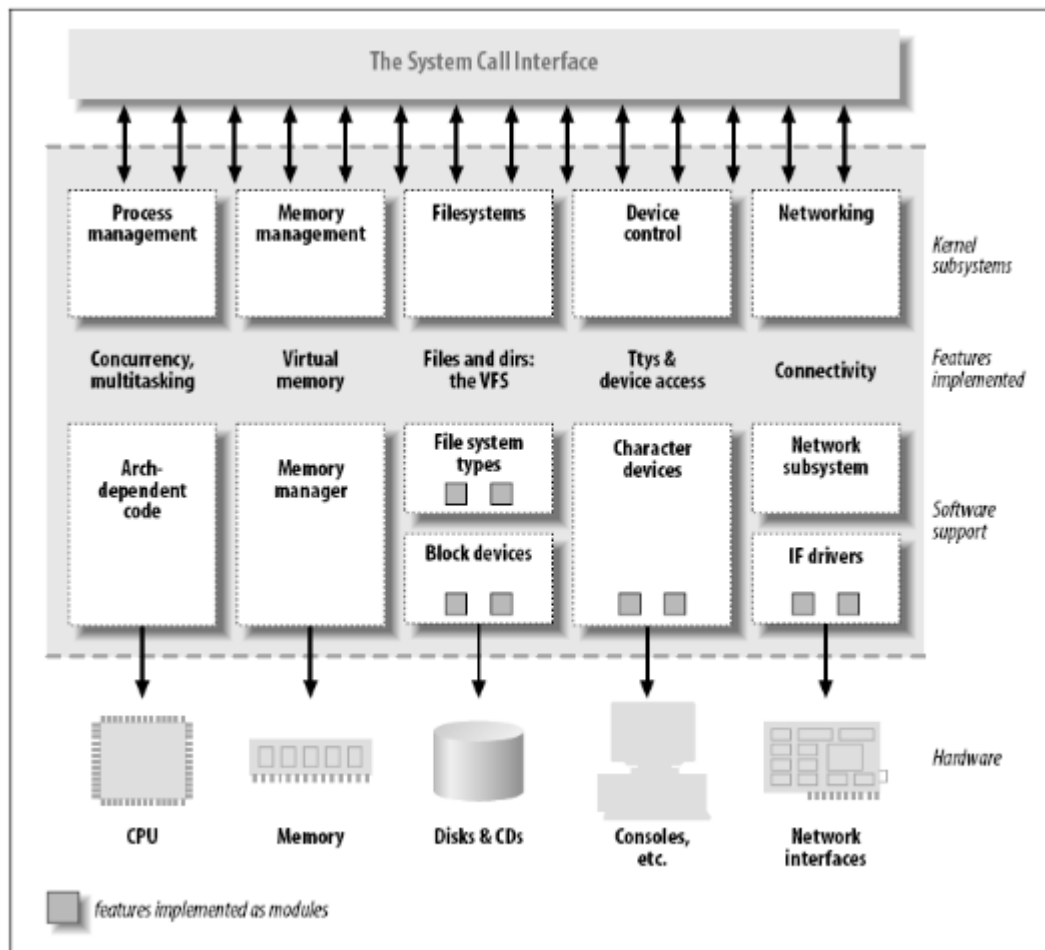
- Aufbau des Linux-Systemkerns am Beispiel Ubuntu und korrekte Zuordnung der Funktionalität
- Kernel-Konfiguration am Beispiel Ubuntu-Linux
- Aufbau, Inhalt und Nutzung von <http://www.kernel.org/>



Allgemeine Struktur von Unix-Systemen

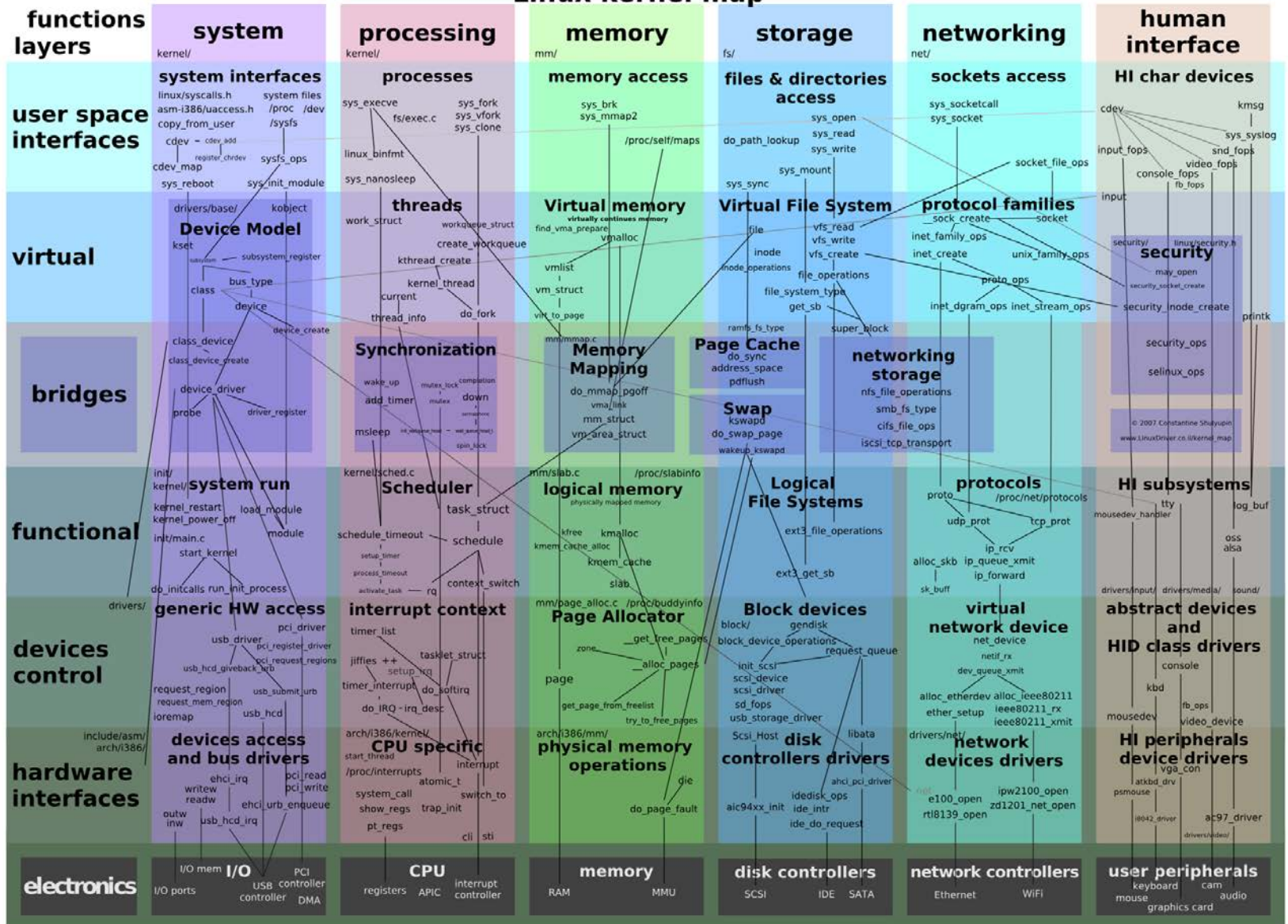


Die Linux-Schichtenarchitektur



Die Linux-Modulstruktur

Linux kernel map



- Der Linux-Kernel ist modular aufgebaut.
- Der Linux-Kernel ermöglicht das dynamische Laden und entfernen von Modulen ohne Neustart des Systems.
- „lsmod“ liefert eine Liste geladener Module auf Basis der Datei /proc/modules (.ko-Datei-Extension).
- „modinfo“ liefert Informationen zu einem spezifischen Modul.
- „depmod -a“ erstellt eine Abhängigkeitsliste der Module in der Datei /lib/modules/version/modules.dep
- „modprobe“ prüft auf Abhängigkeiten und sorgt dann für das Laden der benötigten Module in der richtigen Reihenfolge (kann vom Kernel oder vom Benutzer aufgerufen werden).
- „insmod“ lädt dann die Module in /lib/modules/version/
- „rmmod“ entfernt ein Modul wieder aus dem Kernel.

```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:~$ lsmod | more
Module              Size  Used by
xt_limit            11140  8
xt_tcpudp           11776  10
ipt_LOG             14468  8
ipt_MASQUERADE      11520  0
xt_DSCP             12032  0
ipt_REJECT          11776  1
nf_conntrack_irc    14648  0
nf_conntrack_ftp    17592  0
xt_state            10624  6
binfmt_misc         18572  1
bridge              63904  0
stp                 11140  1 bridge
bnep                 22912  2
input_polldev       12688  0
sbp2                 34700  0
lp                  19588  0
pata_pcmcia          22656  1
snd_hda_intel       557492  4
snd_pcm_oss          52352  0
snd_mixer_oss        24960  1 snd_pcm_oss
snd_pcm              99336  3 snd_hda_intel,snd_pcm_oss
snd_seq_dummy        11524  0
snd_seq_oss          41984  0
snd_seq_midi         15744  0
snd_rawmidi          33920  1 snd_seq_midi
iptables_nat         14724  0
nf_nat               30100  2 ipt_MASQUERADE,iptables_nat
snd_seq_midi_event   16512  2 snd_seq_oss,snd_seq_midi
arc4                 10240  2
nf_conntrack_ipv4    24216  9 iptables_nat,nf_nat
ecb                  11392  2
snd_seq              66272  6 snd_seq_dummy,snd_seq_oss,snd_seq_midi,snd_seq_m
idi_event
nf_conntrack         84752  7 ipt_MASQUERADE,nf_conntrack_irc,nf_conntrack_ftp
,xt_state,iptables_nat,nf_nat,nf_conntrack_ipv4
nf_defrag_ipv4       10496  1 nf_conntrack_ipv4
snd_timer            34064  2 snd_pcm,snd_seq
iwlagn               114820  0
iptables_mangle      11520  0
iwlcore              106496  1 iwlagn
iptables_filter      11392  1
snd_seq_device       16276  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi,snd_rawmi
di,snd_seq
leds_hp_disk         11400  0
pcmcia               47640  1 pata_pcmcia
uvcvideo             69512  0
ip_tables            28304  3 iptables_nat,iptables_mangle,iptables_filter
```

Ismod


```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:~$ depmod -a -v | more
/lib/modules/2.6.28-13-generic/kernel/arch/x86/kernel/cpu/cpufreq/p4-clockmod.ko
needs "speedstep_get_processor_frequency": /lib/modules/2.6.28-13-generic/kerne
l/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/kernel/cpu/cpufreq/p4-clockmod.ko
needs "speedstep_detect_processor": /lib/modules/2.6.28-13-generic/kernel/arch/
x86/kernel/cpu/cpufreq/speedstep-lib.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/crypto/aes-x86_64.ko needs "crypt
o_itab": /lib/modules/2.6.28-13-generic/kernel/crypto/aes_generic.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/crypto/aes-x86_64.ko needs "crypt
o_aes_set_key": /lib/modules/2.6.28-13-generic/kernel/crypto/aes_generic.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/crypto/aes-x86_64.ko needs "crypt
o_fl_tab": /lib/modules/2.6.28-13-generic/kernel/crypto/aes_generic.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/crypto/aes-x86_64.ko needs "crypt
o_il_tab": /lib/modules/2.6.28-13-generic/kernel/crypto/aes_generic.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/crypto/aes-x86_64.ko needs "crypt
o_ft_tab": /lib/modules/2.6.28-13-generic/kernel/crypto/aes_generic.ko
/lib/modules/2.6.28-13-generic/kernel/arch/x86/crypto/twofish-x86_64.ko needs "t
wofish_setkey": /lib/modules/2.6.28-13-generic/kernel/crypto/twofish_common.ko
/lib/modules/2.6.28-13-generic/kernel/fs/nfs_common/nfs_acl.ko needs "xdr_decode
_array2": /lib/modules/2.6.28-13-generic/kernel/net/sunrpc/sunrpc.ko
/lib/modules/2.6.28-13-generic/kernel/fs/nfs_common/nfs_acl.ko needs "xdr_encode
_word": /lib/modules/2.6.28-13-generic/kernel/net/sunrpc/sunrpc.ko
/lib/modules/2.6.28-13-generic/kernel/fs/nfs_common/nfs_acl.ko needs "xdr_encode
_array2": /lib/modules/2.6.28-13-generic/kernel/net/sunrpc/sunrpc.ko
/lib/modules/2.6.28-13-generic/kernel/fs/nfs_common/nfs_acl.ko needs "xdr_decode
_word": /lib/modules/2.6.28-13-generic/kernel/net/sunrpc/sunrpc.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "config_group_init": /
lib/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "config_group_find_ite
m": /lib/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "config_item_get": /li
b/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "configfs_unregister_s
ubsystem": /lib/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "configfs_register_sub
system": /lib/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "config_group_init_typ
e_name": /lib/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "config_item_init_type
_name": /lib/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/dlm/dlm.ko needs "config_item_put": /li
b/modules/2.6.28-13-generic/kernel/fs/configfs/configfs.ko
/lib/modules/2.6.28-13-generic/kernel/fs/fat/vfat.ko needs "fat_dir_empty": /lib
/modules/2.6.28-13-generic/kernel/fs/fat/fat.ko
/lib/modules/2.6.28-13-generic/kernel/fs/fat/vfat.ko needs "fat_time_unix2fat":
/lib/modules/2.6.28-13-generic/kernel/fs/fat/fat.ko
/lib/modules/2.6.28-13-generic/kernel/fs/fat/vfat.ko needs "fat_fs_panic": /lib/
modules/2.6.28-13-generic/kernel/fs/fat/fat.ko
```

depmod

```

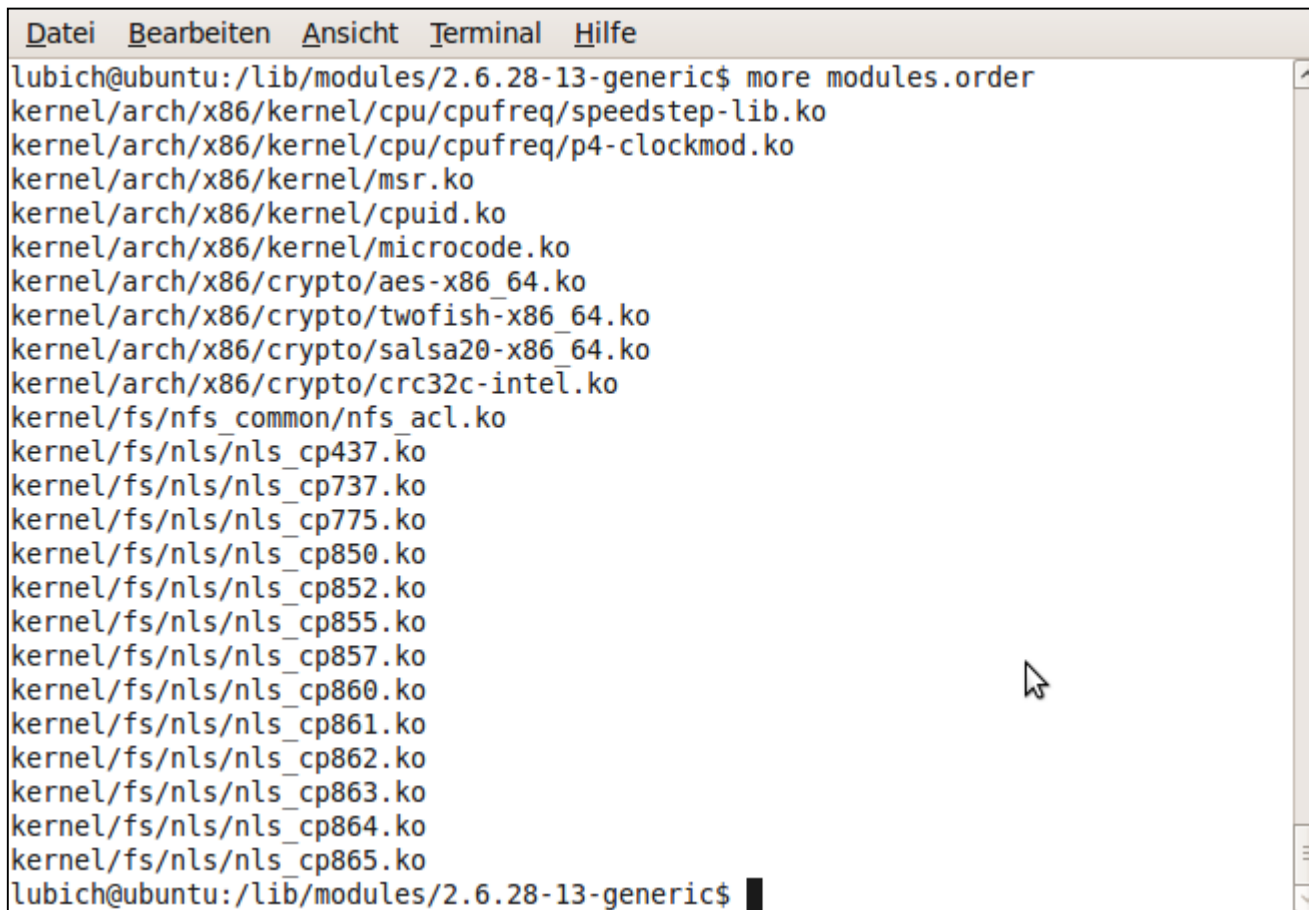
Datei  Bearbeiten  Ansicht  Terminal  Hilfe
lubich@ubuntu:/lib/modules$ ls -ls
insgesamt 16
4 drwxr-xr-x 5 root root 4096 2009-05-15 15:40 2.6.27-11-generic
4 drwxr-xr-x 2 root root 4096 2009-04-29 07:27 2.6.27-7-generic
4 drwxr-xr-x 6 root root 4096 2009-04-29 07:29 2.6.28-11-generic
4 drwxr-xr-x 6 root root 4096 2009-07-02 08:24 2.6.28-13-generic
lubich@ubuntu:/lib/modules$ cd 2.6.28-13-generic
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ ls -ls
insgesamt 2860
0 lrwxrwxrwx 1 root root 40 2009-06-22 20:31 build -> /usr/src/linux-head
ers-2.6.28-13-generic
4 drwxr-xr-x 2 root root 4096 2009-07-02 08:24 initrd
4 drwxr-xr-x 10 root root 4096 2009-06-22 20:30 kernel
452 -rw-r--r-- 1 root root 458099 2009-07-02 08:24 modules.alias
448 -rw-r--r-- 1 root root 451952 2009-07-02 08:24 modules.alias.bin
4 -rw-r--r-- 1 root root 69 2009-07-02 08:24 modules.ccwmap
192 -rw-r--r-- 1 root root 190814 2009-07-02 08:24 modules.dep
284 -rw-r--r-- 1 root root 285228 2009-07-02 08:24 modules.dep.bin
4 -rw-r--r-- 1 root root 887 2009-07-02 08:24 modules.ieee1394map
4 -rw-r--r-- 1 root root 218 2009-07-02 08:24 modules.inputmap
8 -rw-r--r-- 1 root root 7286 2009-07-02 08:24 modules.isapnpmap
4 -rw-r--r-- 1 root root 74 2009-07-02 08:24 modules.ofmap
80 -rw-r--r-- 1 root root 76367 2009-07-01 01:41 modules.order
296 -rw-r--r-- 1 root root 297123 2009-07-02 08:24 modules.pcimap
4 -rw-r--r-- 1 root root 1303 2009-07-02 08:24 modules.seriomap
168 -rw-r--r-- 1 root root 166394 2009-07-02 08:24 modules.symbols
220 -rw-r--r-- 1 root root 217467 2009-07-02 08:24 modules.symbols.bin
680 -rw-r--r-- 1 root root 689618 2009-07-02 08:24 modules.usbmap
4 drwxr-xr-x 3 root root 4096 2009-06-22 20:31 updates
0 drwxr-xr-x 2 root root 360 2009-07-24 10:10 volatile
lubich@ubuntu:/lib/modules/2.6.28-13-generic$

```

```

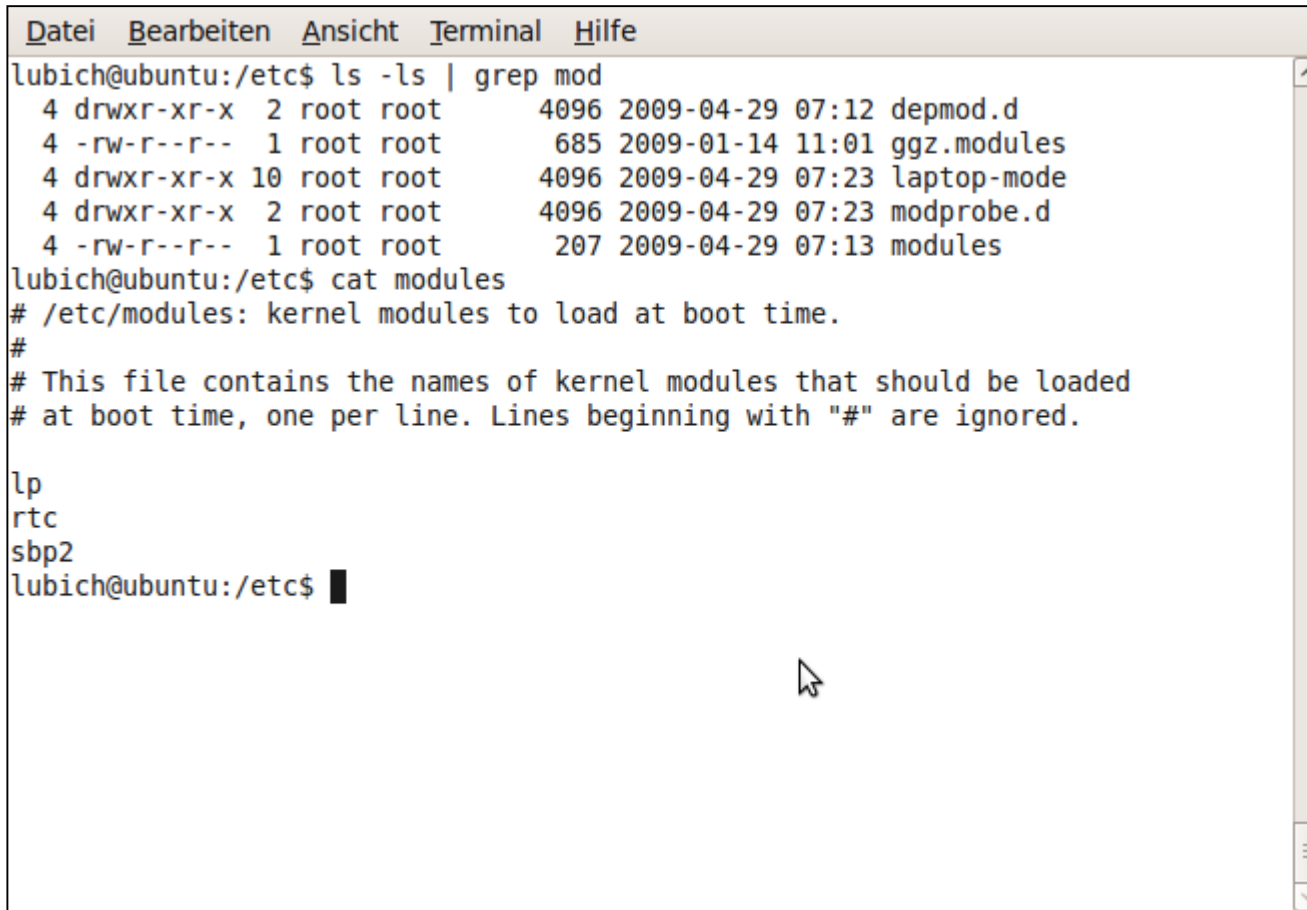
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ file *
build:          symbolic link to `/usr/src/linux-headers-2.6.28-13-generic'
initrd:         directory
kernel:        directory
modules.alias:  ASCII text
modules.alias.bin: data
modules.ccwmap: ASCII text
modules.dep:    ASCII text, with very long lines
modules.dep.bin: data
modules.ieee1394map: ASCII text
modules.inputmap: ASCII text
modules.isapnpmap: ASCII text
modules.ofmap:  ASCII text
modules.order:  ASCII text
modules.pciomap: ASCII C++ program text
modules.seriomap: ASCII text
modules.symbols: ASCII text
modules.symbols.bin: data
modules.usbmap:  ASCII text
updates:        directory
volatile:       directory
lubich@ubuntu:/lib/modules/2.6.28-13-generic$

```



```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ more modules.order
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko
kernel/arch/x86/kernel/cpu/cpufreq/p4-clockmod.ko
kernel/arch/x86/kernel/msr.ko
kernel/arch/x86/kernel/cpuid.ko
kernel/arch/x86/kernel/microcode.ko
kernel/arch/x86/crypto/aes-x86_64.ko
kernel/arch/x86/crypto/twofish-x86_64.ko
kernel/arch/x86/crypto/salsa20-x86_64.ko
kernel/arch/x86/crypto/crc32c-intel.ko
kernel/fs/nfs/common/nfs_acl.ko
kernel/fs/nls/nls_cp437.ko
kernel/fs/nls/nls_cp737.ko
kernel/fs/nls/nls_cp775.ko
kernel/fs/nls/nls_cp850.ko
kernel/fs/nls/nls_cp852.ko
kernel/fs/nls/nls_cp855.ko
kernel/fs/nls/nls_cp857.ko
kernel/fs/nls/nls_cp860.ko
kernel/fs/nls/nls_cp861.ko
kernel/fs/nls/nls_cp862.ko
kernel/fs/nls/nls_cp863.ko
kernel/fs/nls/nls_cp864.ko
kernel/fs/nls/nls_cp865.ko
lubich@ubuntu:/lib/modules/2.6.28-13-generic$
```

Load Modul Konfiguration: /etc/modules



```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:/etc$ ls -ls | grep mod
 4 drwxr-xr-x  2 root root    4096 2009-04-29 07:12 depmod.d
 4 -rw-r--r--  1 root root     685 2009-01-14 11:01 ggz.modules
 4 drwxr-xr-x 10 root root    4096 2009-04-29 07:23 laptop-mode
 4 drwxr-xr-x  2 root root    4096 2009-04-29 07:23 modprobe.d
 4 -rw-r--r--  1 root root     207 2009-04-29 07:13 modules
lubich@ubuntu:/etc$ cat modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

lp
rtc
sbp2
lubich@ubuntu:/etc$
```

Load Modul Konfiguration Blacklist

/etc/modprobe.d/blacklist

```
Datei  Bearbeiten  Ansicht  Terminal  Hilfe
# alias expansion, usually so some other driver will be loaded for the
# device instead.

# evbug is a debug tool that should be loaded explicitly
blacklist evbug

# these drivers are very simple, the HID drivers are usually preferred
blacklist usbmouse
blacklist usbkbd

# replaced by e100
blacklist eeepro100

# replaced by tulip
blacklist de4x5

# causes no end of confusion by creating unexpected network interfaces
blacklist eth1394

# snd_intel8x0m can interfere with snd_intel8x0, doesn't seem to support much
# hardware on its own (Ubuntu bug #2011, #6810)
blacklist snd_intel8x0m

# Conflicts with dvb driver (which is better for handling this device)
blacklist snd_aw2

# causes failure to suspend on HP compaq nc6000 (Ubuntu: #10306)
blacklist i2c_i801

# replaced by p54pci
blacklist prism54

# replaced by b43 and ssb.
blacklist bcm43xx

# most apps now use garmin usb driver directly (Ubuntu: #114565)
blacklist garmin_gps

# replaced by asus-laptop (Ubuntu: #184721)
blacklist asus_acpi

# low-quality, just noise when being used for sound playback, causes
# hangs at desktop session start (Ubuntu: #246969)
```


Load Modul Konfiguration: Reihenfolge

```
lubich@ubuntu: /etc/depmod.d
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:/etc/depmod.d$ ls -ls
insgesamt 4
4 -rw-r--r-- 1 root root 31 2008-10-14 17:52 ubuntu.conf
lubich@ubuntu:/etc/depmod.d$ more ubuntu.conf
search updates ubuntu built-in
lubich@ubuntu:/etc/depmod.d$
```

```
lubich@ubuntu: /lib/modules/2.6.28-13-generic
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:~$ cd /lib/modules
lubich@ubuntu:/lib/modules$ ls
2.6.27-11-generic 2.6.27-7-generic 2.6.28-11-generic 2.6.28-13-generic
lubich@ubuntu:/lib/modules$ cd 2.6.28-13-generic
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ ls
build          modules.ccwmap      modules.isapmap      modules.symbols
initrd         modules.dep         modules.ofmap        modules.symbols.bin
kernel         modules.dep.bin     modules.order        modules.usbmap
modules.alias  modules.ieee1394map modules.pcimap       updates
modules.alias.bin modules.inputmap    modules.seriomap     vmlinux
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ ls -ls updates
insgesamt 4
4 drwxr-xr-x 2 root root 4096 2009-06-22 20:31 dkms
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ ls -ls build
0 lrwxrwxrwx 1 root root 40 2009-06-22 20:31 build -> /usr/src/linux-headers-2.6
.28-13-generic
lubich@ubuntu:/lib/modules/2.6.28-13-generic$ ls -ls kernel
insgesamt 32
4 drwxr-xr-x 3 root root 4096 2009-06-22 20:30 arch
4 drwxr-xr-x 3 root root 4096 2009-07-02 08:24 crypto
4 drwxr-xr-x 53 root root 4096 2009-06-22 20:30 drivers
4 drwxr-xr-x 44 root root 4096 2009-07-02 08:24 fs
4 drwxr-xr-x 5 root root 4096 2009-07-02 08:24 lib
4 drwxr-xr-x 37 root root 4096 2009-06-22 20:30 net
4 drwxr-xr-x 12 root root 4096 2009-07-02 08:24 sound
4 drwxr-xr-x 17 root root 4096 2009-06-22 20:30 ubuntu
lubich@ubuntu:/lib/modules/2.6.28-13-generic$
```

Warum braucht es dynamische Kernel Load Module?

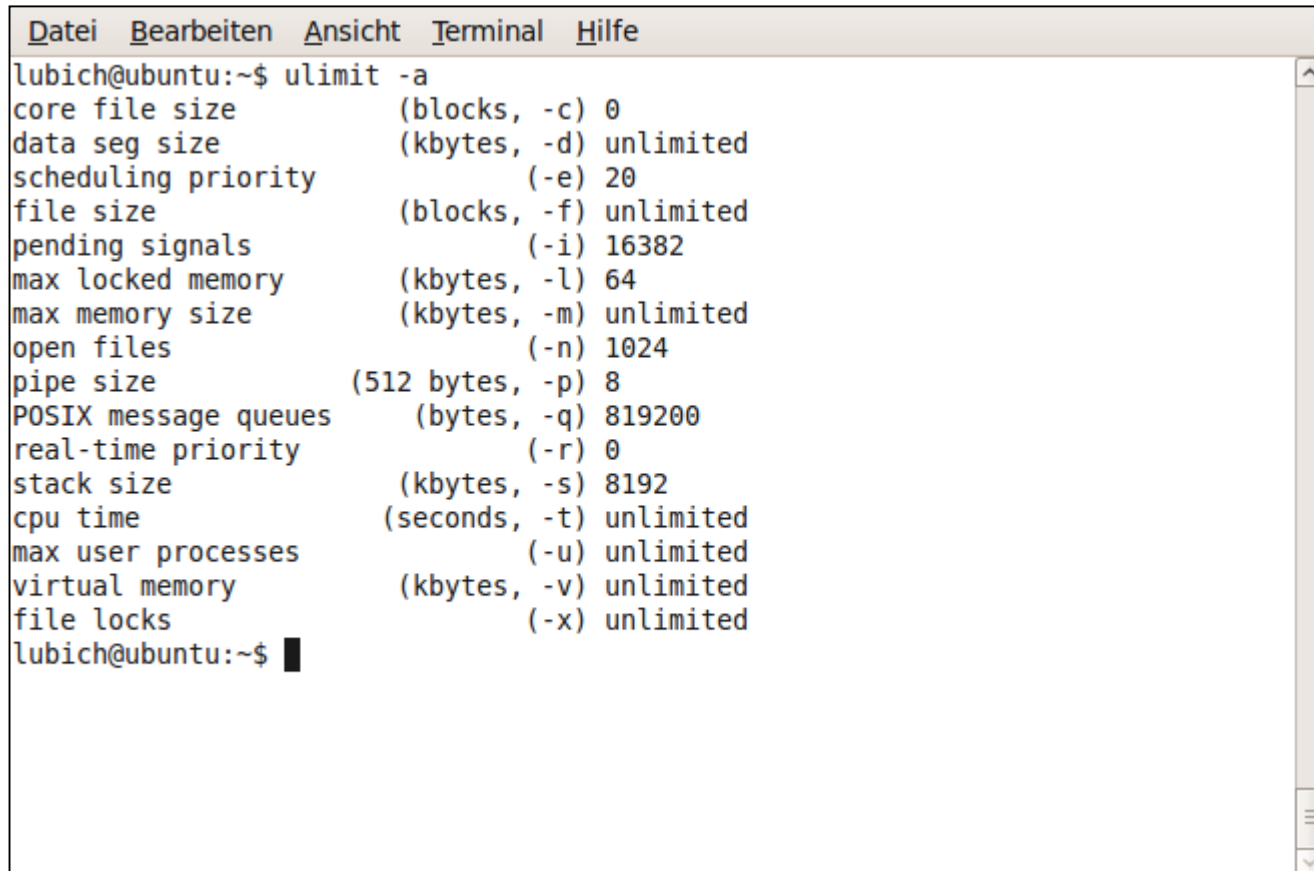
- Generelle Philosophie von Linux als offenes System
- Dynamisches Nachladen von Kernel-Funktionalität (z.B. bei neu erkannten Peripheriegeräten) ohne Reboot
- Austausch fehlerhafter Software im Betrieb
- Rasche Entwicklung am laufenden System (ggf. ohne Reboot, z.B. mit KernelCare, kGraft, Kpatch)

Konsequenzen von dynamischen Kernel Load Modulen

- „Use at you own risk“: Kernel Load Module umgehen den Schutz der System Call Schnittstelle und können direkt mit sensiblen Kernel-Datenstrukturen interagieren – Probleme führen dabei oft zum Systemabsturz.
- „tainting“: Der Kernel wird durch von Benutzern geschriebene Load Module „verunreinigt“, d.h. diese Module können direkte Auswirkungen auf den Kernel haben, die die Kernel-Entwickler nicht mehr im Griff haben und nicht debuggen können bzw. wollen.
- Fragmentierung: der Linux-Kernel wird unfragmentiert in den Speicher geladen, das Laden und Entfernen von Kernel Load Modulen zur Laufzeit führt zur fragmentierten Speicherung und damit zu einem möglichen Performance-Verlust.
- Sicherheit: Kernel Load Module haben direkten Zugriff auf die Datenstrukturen des Kernels (z.B. das Stück Speicher, welches gerade temporär mein Login-Passwort enthält). Man muss also fremde (und eigene) Kernel Load Module genau auf Nebeneffekte überprüfen.
- API: Linux hat kein stabiles Application Programmer's Interface für Kernel Load Module – bei neuen Versionen kann dies zu Kompatibilitätsproblemen zur Übersetzungs- und zur Laufzeit führen.

Weitere Modifikationen am Kernel

- Veränderung von systemweiten oder benutzerbezogenen Verbrauchslimiten
 - Veränderung von Systemparametern (temporär oder permanent)
- Auswirkungen auf Stabilität und Performance



```
lubich@ubuntu:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority     (-e) 20
file size                (blocks, -f) unlimited
pending signals         (-i) 16382
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files              (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues    (bytes, -q) 819200
real-time priority      (-r) 0
stack size              (kbytes, -s) 8192
cpu time                (seconds, -t) unlimited
max user processes      (-u) unlimited
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
lubich@ubuntu:~$
```

Kernel Konfiguration & Tuning: sysctl

```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:~$ sysctl -a | sort | more
error: permission denied on key 'kernel.cad_pid'
error: permission denied on key 'fs.binfmt_misc.register'
error: permission denied on key 'dev.parport.parport0.autoprobe'
error: permission denied on key 'dev.parport.parport0.autoprobe0'
error: permission denied on key 'dev.parport.parport0.autoprobe1'
error: permission denied on key 'dev.parport.parport0.autoprobe2'
error: permission denied on key 'dev.parport.parport0.autoprobe3'
error: permission denied on key 'net.ipv4.route.flush'
error: permission denied on key 'net.ipv6.route.flush'
abi.vsyscall32 = 1
crypto.fips_enabled = 0
debug.exception-trace = 1
dev.cdrom.autoclose = 1
dev.cdrom.autoeject = 0
dev.cdrom.check_media = 0
dev.cdrom.debug = 0
dev.cdrom.info =
dev.cdrom.info =
dev.cdrom.info =
dev.cdrom.info = Can change speed:      1
dev.cdrom.info = Can close tray:          1
dev.cdrom.info = Can lock tray:           1
dev.cdrom.info = Can open tray:           1
dev.cdrom.info = Can play audio:          1
dev.cdrom.info = Can read DVD:            1
dev.cdrom.info = Can read MCN:            1
dev.cdrom.info = Can read MRW:            1
dev.cdrom.info = Can read multisession: 1
dev.cdrom.info = Can select disk:         0
dev.cdrom.info = Can write CD-R:          1
dev.cdrom.info = Can write CD-RW:         1
dev.cdrom.info = Can write DVD-R:         1
dev.cdrom.info = Can write DVD-RAM:       1
dev.cdrom.info = Can write MRW:           1
dev.cdrom.info = Can write RAM:           1
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info = drive name:              sr0
dev.cdrom.info = drive # of slots:        1
dev.cdrom.info = drive speed:             24
dev.cdrom.info = Reports media changed: 1
dev.cdrom.lock = 1
dev.hpet.max-user-freq = 64
dev.mac_hid.mouse_button2_keycode = 97
dev.mac_hid.mouse_button3_keycode = 100
dev.mac_hid.mouse_button_emulation = 0
dev.parport.default.spintime = 500
dev.parport.default.timeslice = 200
lubich@ubuntu:~$
```

```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:/etc$ more sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 4 4 1 7

#####3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# This disables TCP Window Scaling (http://lkml.org/lkml/2008/2/5/167),
# and is not recommended.
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Ignore ICMP broadcasts
#net.ipv4.icmp_echo_ignore_broadcasts = 1
#
# Ignore bogus ICMP errors
#net.ipv4.icmp_ignore_bogus_error_responses = 1
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
```

Kernel Konfiguration & Tuning: sysctl.conf

Kernel Konfig. & Tuning: sysctl.d

```
Datei Bearbeiten Ansicht Terminal Hilfe
lubich@ubuntu:/etc/sysctl.d$ ls -ls
insgesamt 20
4 -rw-r--r-- 1 root root 77 2008-10-27 12:17 10-console-messages.conf
4 -rw-r--r-- 1 root root 509 2009-03-18 23:35 10-network-security.conf
4 -rw-r--r-- 1 root root 70 2008-10-15 10:16 30-tracker.conf
4 -rw-r--r-- 1 root root 107 2009-03-30 15:57 30-wine.conf
4 -rw-r--r-- 1 root root 450 2009-03-18 23:35 README
lubich@ubuntu:/etc/sysctl.d$ cat README
This directory contains settings similar to those found in /etc/sysctl.conf.
In general, files in the 10-*.conf range come from the procs package and
serve as system defaults. Other packages install their files in the
30-*.conf range, to override system defaults. End-users can use 60-*.conf
and above, or use /etc/sysctl.conf directly, which overrides anything in
this directory.

After making any changes, please run "invoke-rc.d procs start".
lubich@ubuntu:/etc/sysctl.d$ cat 10-console-messages.conf

# the following stops low-level messages on console
kernel.printk = 4 4 1 7
lubich@ubuntu:/etc/sysctl.d$ cat 10-network-security.conf

# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks.
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1

# Turn on SYN-flood protections. Starting with 2.6.26, there is no loss
# of TCP functionality/features under normal conditions. When flood
# protections kick in under high unanswered-SYN load, the system
# should remain more stable, with a trade off of some loss of TCP
# functionality/features (e.g. TCP Window scaling).
net.ipv4.tcp_syncookies=1
lubich@ubuntu:/etc/sysctl.d$ █
```


Linux-Kernel: Mehr hauptberufliche Entwickler

In der inzwischen fünften Neuauflage der Studie „Linux Kernel Development – How Fast It is Going, Who is Doing It, What They are Doing and Who is Sponsoring It“ haben Jonathan Corbet (LWN.net) sowie die bei der Linux Foundation angestellten Greg Kroh-Hartman und Amanda McPherson die Beiträge zur Kernel-Entwicklung untersucht. Sie konzentrierten sich bei ihrer jetzt auf der amerikanischen LinuxCon in New Orleans vorgestellten Statistik vor allem auf die Kernel-Versionen zwischen 3.3 und 3.10.

Texas Instruments in der Studie von 2012 4,4 Prozent aller Patches; der Anteil hat sich 2013 auf 11 Prozent erhöht. Die Zahlen zeigen, dass Linux in der Industrie angekommen ist und widerlegen deutlich das immer noch gern verwendete Argument vom Hobbyisten-Betriebssystem.

Fast parallel zur Studie veröffentlichte Linus Torvalds die derzeit aktuelle Kernel-Version 3.11. Als Erinnerung an die zwanzig Jahre zuvor erschienene, als erste von Haus aus TCP/IP-fähige Windows-Version änderte er bei der Freigabe den ursprünglichen Codenamen von „Unicycling Gorilla“ auf „Li-

Linux-Kernel-Statistik

Version	Dateien	Code-Zeilen	Patches	Entwickler	Firmen
3.0	36 788	14 651 135	9153	1131	191
3.1	37 095	14 776 002	8693	1168	189
3.2	37 626	15 004 006	11 780	1316	231
3.3	38 091	15 171 607	10 550	1247	233
3.4	38 573	15 389 393	10 889	1286	245
3.5	39 101	15 601 911	10 957	1195	242
3.6	39 738	15 873 569	10 247	1224	298
3.7	40 912	16 197 233	11 990	1280	228
3.8	41 532	16 422 416	12 394	1258	241
3.9	42 435	16 692 421	11 910	1388	263
3.10	43 029	16 961 031	13 367	1392	243

Quelle: Linux Foundation

Quelle: iX, 10/2013

- The Linux Kernel Archives:
<http://www.kernel.org/> → Quellcode-Sammlung des Linux Kernel Codes
- The Linux Documentation Project:
<http://tldp.org/> → diverse Linux-Dokumentationen, inkl. Kernel-Module etc.

Zusammenfassung der Lektion 9 und Hausaufgabe

- Grundlagen der Virtualisierung
- Grundsätzlicher Aufbau des Linux-Systemkerns und Identifikation der funktionalen Elemente und Aufgaben.
- Konfigurationsmöglichkeiten des Linux-Kernels und Konfigurationsänderungen bzw. -erweiterungen.
- Hausaufgabe:
 - Repetieren Sie den Stoff dieser Lektion.
 - Studieren Sie das Material unter den Link „[https://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](https://de.wikipedia.org/wiki/Virtualisierung_(Informatik))»
 - Studieren Sie das Dokument „9-ibm-linux-kernel.pdf“
 - Beenden Sie die Installation von Linux-MINT aus der Übung – unter <https://www.osboxes.org/virtualbox-images/> finden Sie bei weitere virtuelle Maschinen für VirtualBox