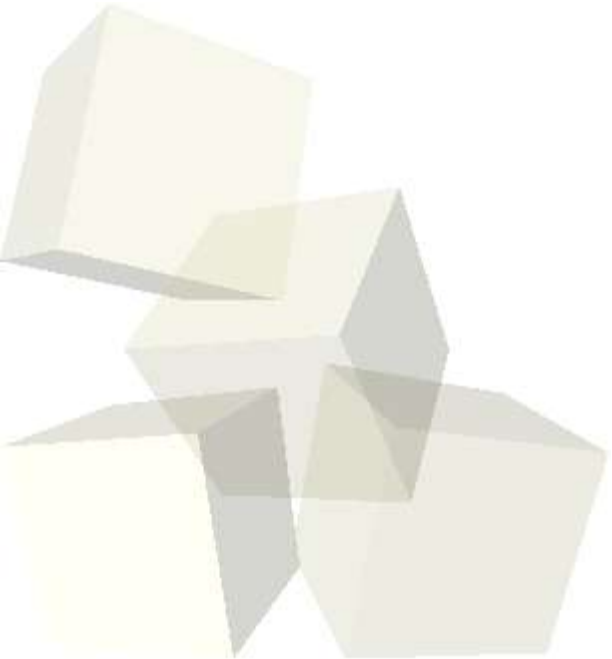
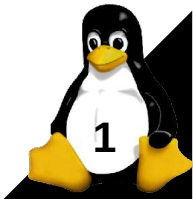
Linux Systemadministration

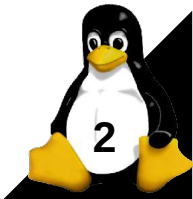
DI (FH) Descher Marco



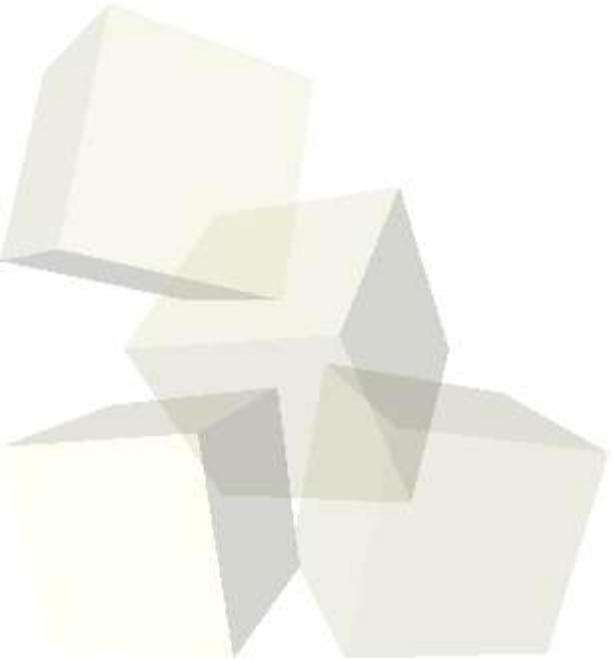


Programm

- Einführung
- Grundkonzepte
- Vom Systemstart zur Shell
- Arbeiten mit der Shell
- Prozessmanagement / Logmanagement
- Benutzerverwaltung
- Grundlegende Netzwerkkonzepte
- Netzwerk Dateisysteme
- Systemsicherheit
- Sonstiges bzw. Fragen



Einführung




Linus Torvalds

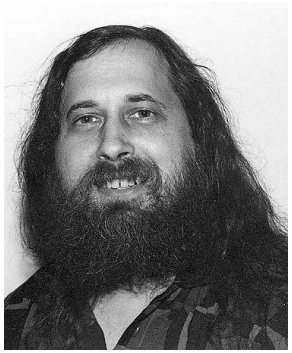
Linux: Linus' Unix (zu Beginn “Freax”)

- 1991 Linus Torvalds postet auf comp.os.minix
  “Hello everybody out there using minix...”
- 1993 Erste deutsche Linux Distribution (DLD)
- 1994 *Kernel 1.0* erscheint
- 1995 Portierungen auf Sun Sparc und Digital Alpha
- 1996 *Kernel 2.0* bringt SMP (Symmetric Multi-Processing)
- 1997 Anstoß für Enterprise Features (HA usw.)
- 1998 Beowulf Linux Cluster stellt Weltrekord auf
- 1999 Pinguine auf der CeBIT, SAP R/3, IBM Support
- 2000 Höhepunkt Linux-Hype mit Börsengängen
- 2001 Dümpelnde Weltwirtschaft, Microsoft-FUD

Info: http://www.linux-magazin.de/Artikel/ausgabe/2001/10/10-Jahre-Linux/10-Jahre.html

GNU/Linux

- Linux *nur* Kern des Betriebssystems
- GNU-Tools stellen große Teile fertiger Linux Distributionen (zB.: gcc, bash, emacs, ...)
- Korrekte Bezeichnung daher eigentlich *GNU/Linux*
- Richard Stallman Initiator der GNU/FSF Initiative



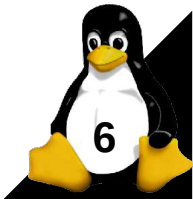
Richard Stallman

Info: http://www.gnu.org

Distributionen

- GNU/Linux für jeden aus Einzelpaketen selbst zusammenbaubar -> Projekte zB.: www.linuxfromscratch.org
- Bieten fertig assemblierte GNU/Linux Systeme zum Teil mit Handbüchern und Support-Verträgen
- Unmenge an Distributionen vorhanden
  (Kurzüberblick auf www.linuxiso.org möglich)
- Großteil für Einsatz in KMUs *ungeeignet*, da fehlender Firmenbackground bzw. kein direkter Support

- Zwei "Business-Grade Linux Anbieter" mit Support:
  - RedHat Linux (www.redhat.de)
    - RedHat Enterprise Linux Edition
  - SuSE Linux (www.suse.de)
    - SuSE Enterprise Server

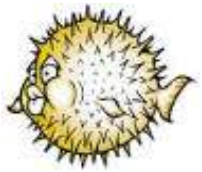


Exkurs: BSD

BSD: Berkeley Software Distribution

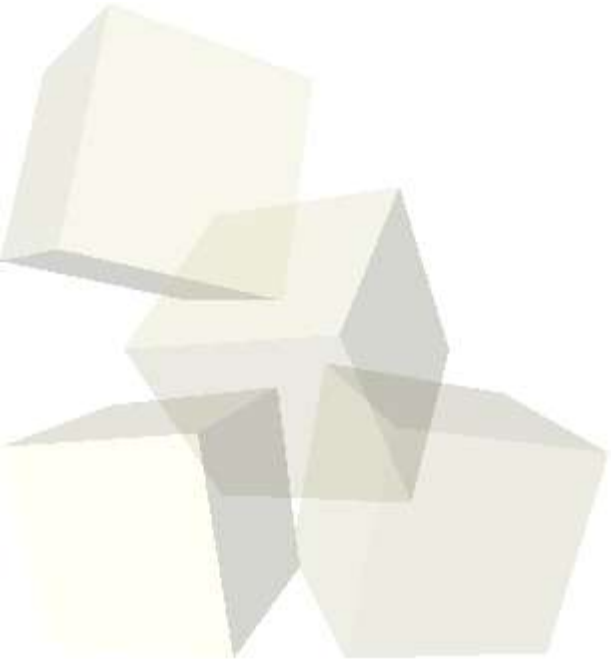
- "BSD hat als Ziel das Ur-Unix auf den PC zu portieren, Linux das Ziel ein Unix-System für den PC zu schreiben"
- Open Source Implementation eines komplett UNIX-kompatiblen Betriebssystems, entwickelt an der University of Californa, Berkeley

- Drei große BSD Unterarten:
  - FreeBSD
    - → *Fokus:* Maximale Stabilität und Robustheit für Desktop und Server
  - NetBSD
    - → *Fokus:* Soll auf soviel Plattformen wie möglich verfügbar sein
  - OpenBSD
    - → *Fokus:* Maximale Sicherheit

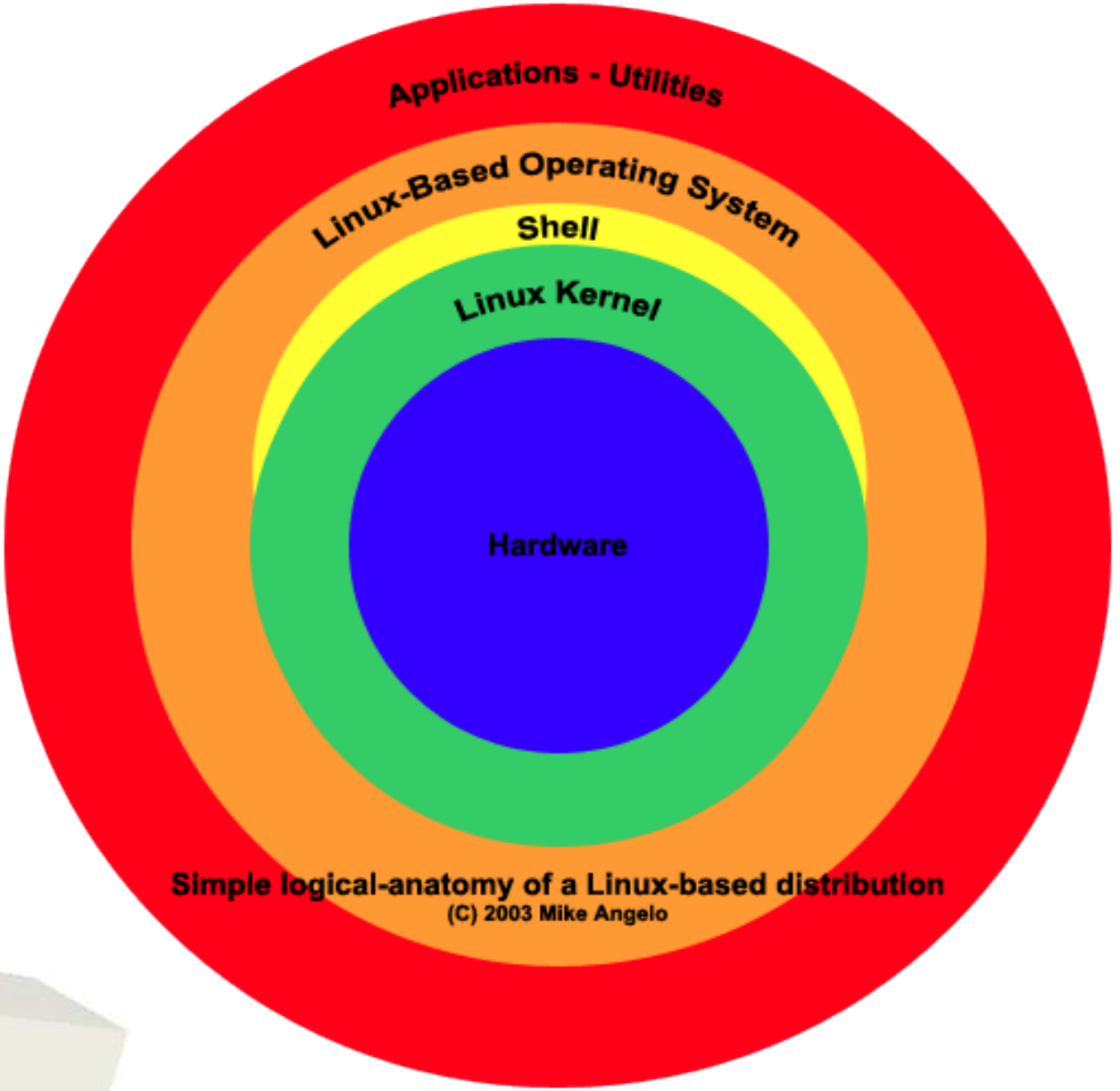


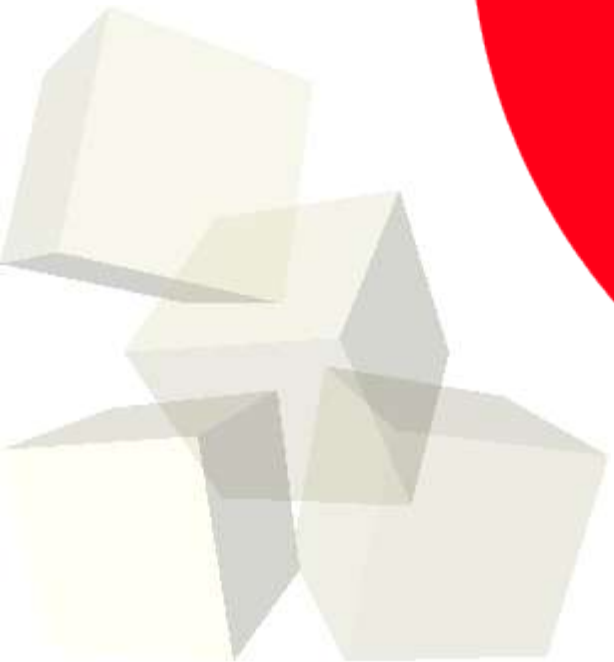
Info: http://www.bsd.org, http://www.over-yonder.net/~fullermd/rants/bsd4linux/bsd4linux1.php

Grundkonzepte



Systemüberblick

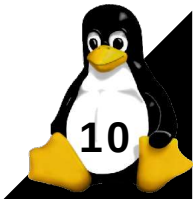




Linux itself: Der Kernel

- Basissystem zwischen Hardware und höherer Software
- Macht Hardware für die auf dem System laufenden Anwendungen durch einheitliche API verfügbar
- Ist kein komplettes Betriebssystem, da ohne weitere Programme relative nutzlos
- zB zuständig für Speicherverwaltung, Prozessverwaltung, Multitasking, I/O Operationen, ...

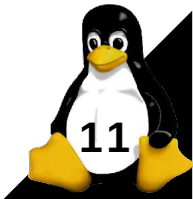
Info: http://de.wikipedia.org/wiki/Linux-Kernel, http://www.kernelnewbies.org



Versionen und Geschmacksrichtungen

- Kernel Versionen:
  - 2.6 - aktuellste Version (Einsatz eher auf Desktops)
  - 2.4 - stabile Business-Version (Einsatz vor allem auf Servern)
  - 2.2 - kaum noch im Einsatz
  - 2.0 - kaum noch im Einsatz
- Geschmacksrichtungen:
  - *Vanilla:* Original Kernel, am meisten getestet
  - *-ac, -ck, -dj:* Von Kernel-Hackern modifizierte original Kernel mit verschiedenen Zielen (neue Treiber, Security..)
  - *Distributions-Kernel:* Von den einzelnen Distributoren angepasste Kernel (an Benutzerbedürfnisse angepasst)

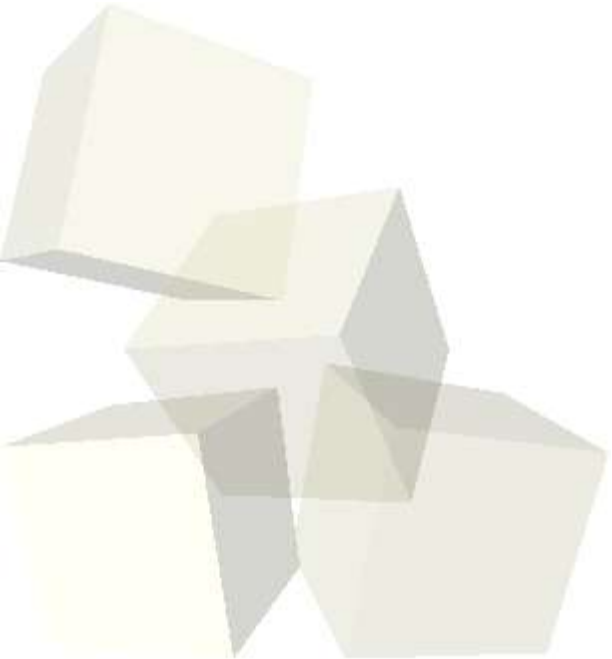
Info: http://www.kernel.org, http://www.kernelnewbies.org

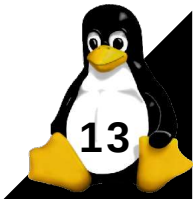


Über welchen root reden wir jetzt?

- **root** ist der höchste user eines Linux-Systems, Zugriffsrechte sind für ihn nicht gültig, da **root** das System ist.
  - Die Tag für Tag Arbeit sollte auf keinen Fall als root durchgeführt werden, da ein falsch angewendeter Befehl fatale Folgen haben kann
- **/** (sprich root) ist das höchste *Verzeichnis*, sozusagen der Start-punkt des Systems
- **/root** ist das Home-Verzeichnis des users root

Vom Starten zur Shell





Zeitlinie Bootprozess

Timeline of the LINUX BOOT PROCESS

BOOT | SHUT DOWN | REBOOT

1 Hardware (BIOS chip on motherboard) reads master boot record.

2 LILO loads.

3 Kernel loads.

4 Root file system is mounted.

5 Start init, the "grandfather" of all processes.

6 Read /etc/inittab.

7 Run all scripts pointed to from rcS (via a line in inittab): "/etc/init.d/rcS".

/etc/rcS.d/S03localnet -> ../init.d/localnet

/etc/rcS.d/S05checkfs -> ../init.d/checkfs

/etc/rcS.d/S10mountfs -> ../init.d/mountfs

/etc/rcS.d/S20setclock -> ../init.d/setclock

8 Run all scripts pointed to from the default runlevel, which is 3 (via a line in inittab): "/etc/init.d/rc 3".

/etc/rc3.d/S03sysklogd -> ../init.d/sysklogd

/etc/rc3.d/S10ethnet -> ../init.d/ethnet

9 Agetty brings up login prompt (via a line in inittab): "/sbin/agetty tty1 9600". The system is ready.

1 Run all scripts pointed to from runlevel 0 (via a line in inittab): "/etc/init.d/rc 0".

/etc/rc0.d/K90syslogd -> ../init.d/sysklogd

/etc/rc0.d/S80sendsignals -> ../init.d/sendsignals

/etc/rc0.d/S90umountfs -> ../init.d/umountfs

/etc/rc0.d/S99halt -> ../init.d/halt

1 Run all scripts pointed to from runlevel 6 (via a line in inittab): "/etc/init.d/rc 6".

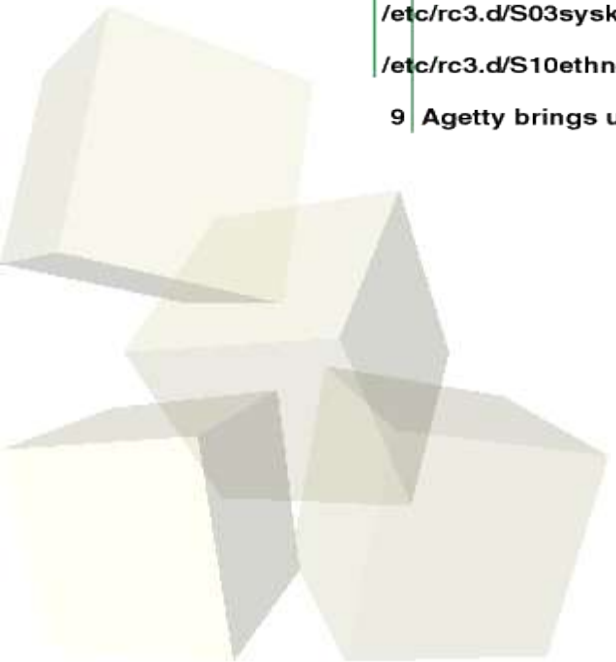
/etc/rc6.d/K90sysklogd -> ../init.d/sysklogd

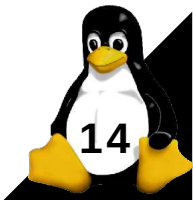
/etc/rc6.d/S80sendsignals -> ../init.d/sendsignals

/etc/rc6.d/S90umountfs -> ../init.d/umountfs

/etc/rc6.d/S99reboot -> ../init.d/reboot

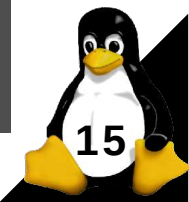
Quelle: http://www.ccoss.org/tutorials/lfs/Linux_from_Scratch_A_Tour.htm





Der Bootloader

- *Paradox*: Um das Betriebssystem in den Speicher laden zu können, braucht es bereits ein Betriebssystem
- *Lösung*: Bootloader, nur für diesen Zweck angepasstes "Betriebssystem"
- Bietet Möglichkeit zur Auswahl der Boot-Quelle (zB.: Festplatte, Netzwerk, Diskette, CD-ROM...)
- Zwei führende Implementationen unter Linux
  - **GRUB** (GNU Grand Unified Boot Loader)
    aktuell gebräuchliche Version – Standard Boot Loader in allen Linux Distributionen
    http://www.gnu.org/software/grub/
  - **LILO** (Linux Loader)
    http://lilo.go.dyndns.org/pub/linux/lilo/
- Lädt den Kernel in den Speicher
- Übergibt Kernel Startargumente
- Führt den Kernel aus, und übergibt ihm die Kontrolle

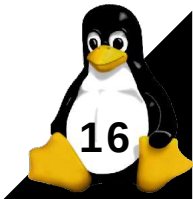


Der Kernel übernimmt die Kontrolle

- Kernel konfiguriert sich auf das System (unter Berücksichtigung der ihm gegebenen Parameter), und startet nach dem mounten des root filesystems (/) den **init** prozess.

```
Linux version 2.4.27-pre2 (root@linux-cluster) (gcc version 3.3.2 20031218 (Gentoo Linux 3.3.2-
r5, propolice-3.3-7)) #6 Mon May 10 10:05:08 CEST 2004
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
 BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000e0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000001fe40000 (usable)
 BIOS-e820: 000000001fe40000 - 000000001fe50000 (ACPI data)
 BIOS-e820: 000000001fe50000 - 000000001ff00000 (ACPI NVS)
510MB LOWMEM available.
On node 0 totalpages: 130624
zone(0): 4096 pages.
zone(1): 126528 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/hda3 noapic
Initializing CPU#0
Detected 2666.815 MHz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 5321.52 BogoMIPS
Memory: 514192k/522496k available (1453k kernel code, 7916k reserved, 505k data, 88k init, 0k
highmem)
Dentry cache hash table entries: 65536 (order: 7, 524288 bytes)
Inode cache hash table entries: 32768 (order: 6, 262144 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 32768 (order: 5, 131072 bytes)
Page-cache hash table entries: 131072 (order: 7, 524288 bytes)
CPU: Trace cache: 12K uops, L1 D cache: 8K
CPU: L2 cache: 512K
Intel machine check architecture supported.
Intel machine check reporting enabled on CPU#0.
CPU:     After generic, caps: bfebfbff 00000000 00000000 00000000
CPU:          Common caps: bfebfbff 00000000 00000000 00000000
CPU: Intel(R) Pentium(R) 4 CPU 2.66GHz stepping 09
Enabling fast FPU save and restore... done.
Enabling unmasked SIMD FPU exception support... done.
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
mtrr: v1.40 (20010327) Richard Gooch (rgooch@atnf.csiro.au)
mtrr: detected mtrr type: Intel
PCI: PCI BIOS revision 2.10 entry at 0xf0031, last bus=1
PCI: Using configuration type 1
PCI: Probing PCI hardware
PCI: Probing PCI hardware (bus 00)
PCI: Probing PCI hardware (bus 00)
PCI: Ignoring BAR0-3 of IDE controller 00:1f.1
Transparent bridge - Intel Corp. 82801BA/CA/DB/EB PCI Bridge
PCI: Using IRQ router PIIX/ICH [8086/24c0] at 00:1f.0
PCI: Found IRQ 10 for device 00:1f.1
PCI: Sharing IRQ 10 with 00:1d.2
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
apm: BIOS not found.
Starting kswapd
Journalled Block Device driver loaded
devfs: v1.12c (20020818) Richard Gooch (rgooch@atnf.csiro.au)
devfs: boot_options: 0x1
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ SERIAL_PCI enabled
Uniform Multi-Platform E-IDE driver Revision: 7.00beta4-2.4
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
ICH4: IDE controller at PCI slot 00:1f.1
.
.
.
.
.
```

Auszug der Kernel Bootup-Messages, generiert mitels dem **dmesg** Befehl.



Der *init* Prozess / Runlevels

- Falls nicht als Parameter anders angegeben, startet der Kernel den init Prozess, und übergibt diesem die Kontrolle

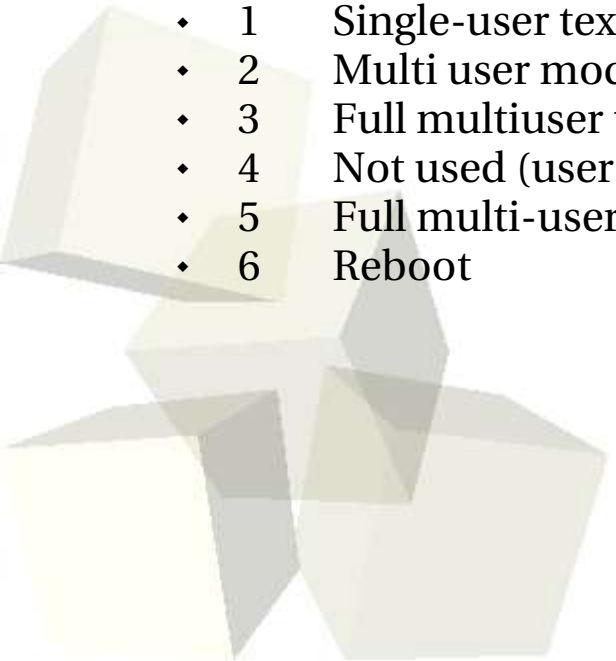
  Auszug Kernel: `init/main.c: run_init_process("/sbin/init");`

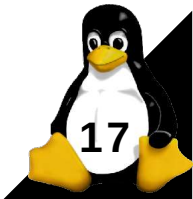
- Der init Prozess startet nun alle weiterführenden Programme wie sie laut `/etc/inittab` definiert sind, von hier aus wird in den Default-Runlevel gewechselt, und die darin definierten Programme gestartet

- **Runlevels** sind Zustände in denen sich das System befinden kann.

  Beispiele für Runlevels, sie können je nach Distribution variieren:

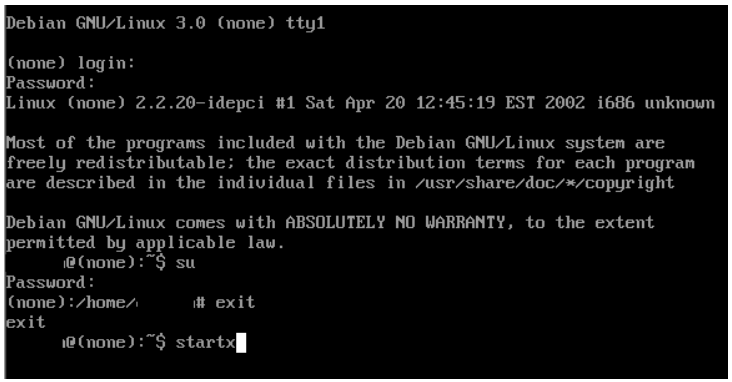
  - 0 Halt
  - 1 Single-user text mode
  - 2 Multi user mode networking disabled
  - 3 Full multiuser text mode
  - 4 Not used (user definable)
  - 5 Full multi-user graphical mode (with an X-based login screen)
  - 6 Reboot

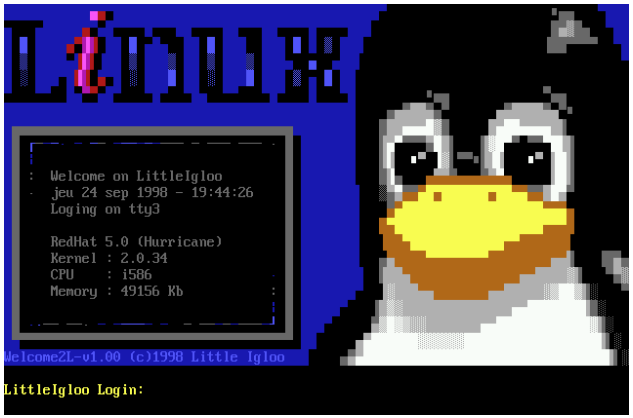




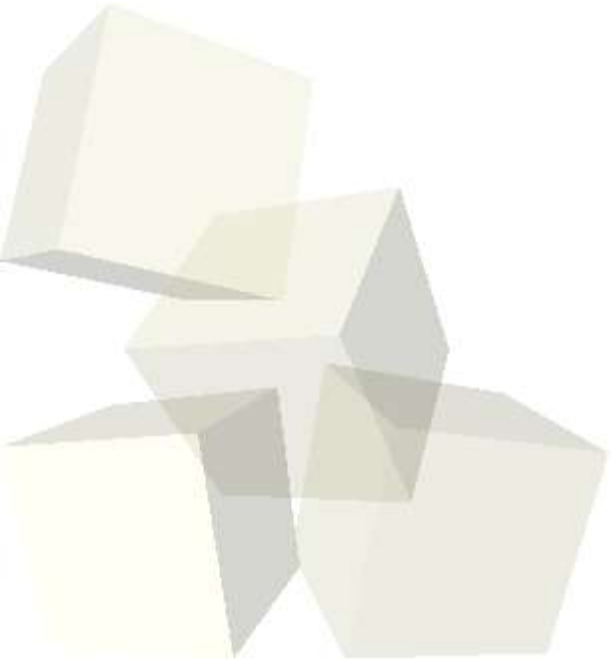
Das Login

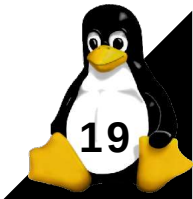
- Das Login stellt den Einstiegspunkt für den User in das System dar, hier authentifiziert sich ein User
- Linux ist *case-sensitiv*, was bedeutet das zum Beispiel die Anmeldung als Root zu einem anderen User führen würde, als die Anmeldung als root
- Der root-account sollte **nur zur Systemwartung**, und auf keinen Fall zur täglichen Arbeit benutzt werden!





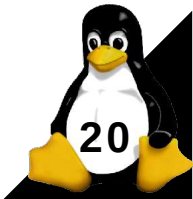
Arbeiten in der Shell





Learn how to learn

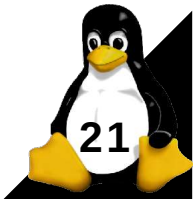
- Es gibt drei Quellen, die bei der täglichen Arbeit unter Linux wichtige Hilfe bieten, und bei richtiger Anwendung so gut wie alle Probleme lösen
  - **Linux Manpages**
    Die wichtigste Quelle für Information. Diese Manpages existieren normalerweise für sämtliche Befehle die man in einem *nix-System finden kann.
    Anwendung: `man Manpage` (zB.: `man login` – Liefert Manualpage der login shell)
  - **HOWTOs / Guides / miniHOWTOs**
    Die Homepage des Linux Documentation Projects (http://www.tldp.org) bietet einen großen Fundus an sogenannten HOWTOs. Diese HOWTOs behandeln konkrete Thematiken, und sind eine wichtige Quelle für Informationen, die sich nicht mit der Manpage eines einzelnen Programmes abtun lassen.
    zB.: Linux IP Masquerade HOWTO
  - **FAQs und Dokumentation auf den Projektseiten**
    Die Homepages einzelner Open Source Projekte bieten meist extensive Dokumentationen an die ebenfalls eine wichtige Hilfsquelle darstellen.



Die Shell

- Linux unterstützt zwei Arten der Kommandoeingabe, diese werden je nach ausgewähltem Default Runlevel gestartet:
  - *Text Kommandozeilen Shells*
    - zB.: bash, tcsh, sh, ...
      Verhältnissmässig schwer zu erlernen, aber unerlässlich um effektiv ein *nix System bedienen bzw. administrieren zu können
  - *Grafische Anwender Interfaces (GUI)*
    - zB.: KDE, GNOME, Xfce, ...
      Einfach in der Bedienung, erlauben aber kaum komplexe Steuerungen

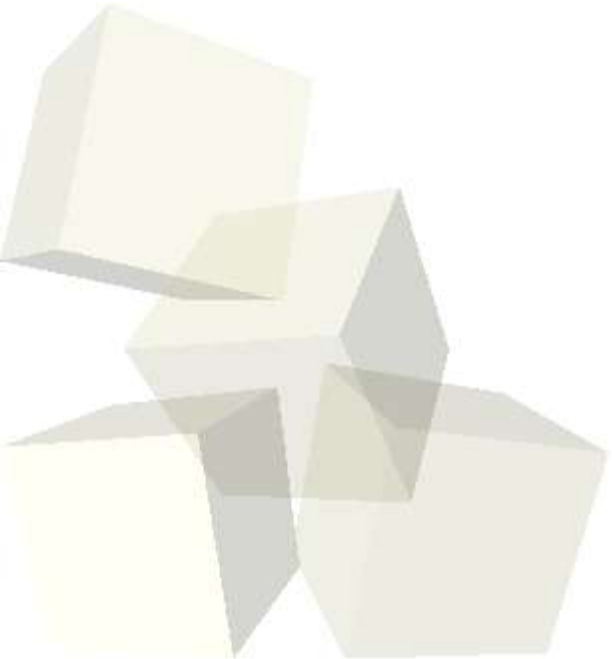
- *bash* (Bourne again shell) mittlerweile wichtigste Shell in den meisten Linux-Distributionen, aufgrund ihrer Mächtigkeit und trotzdem relativ einfachen Bedienung



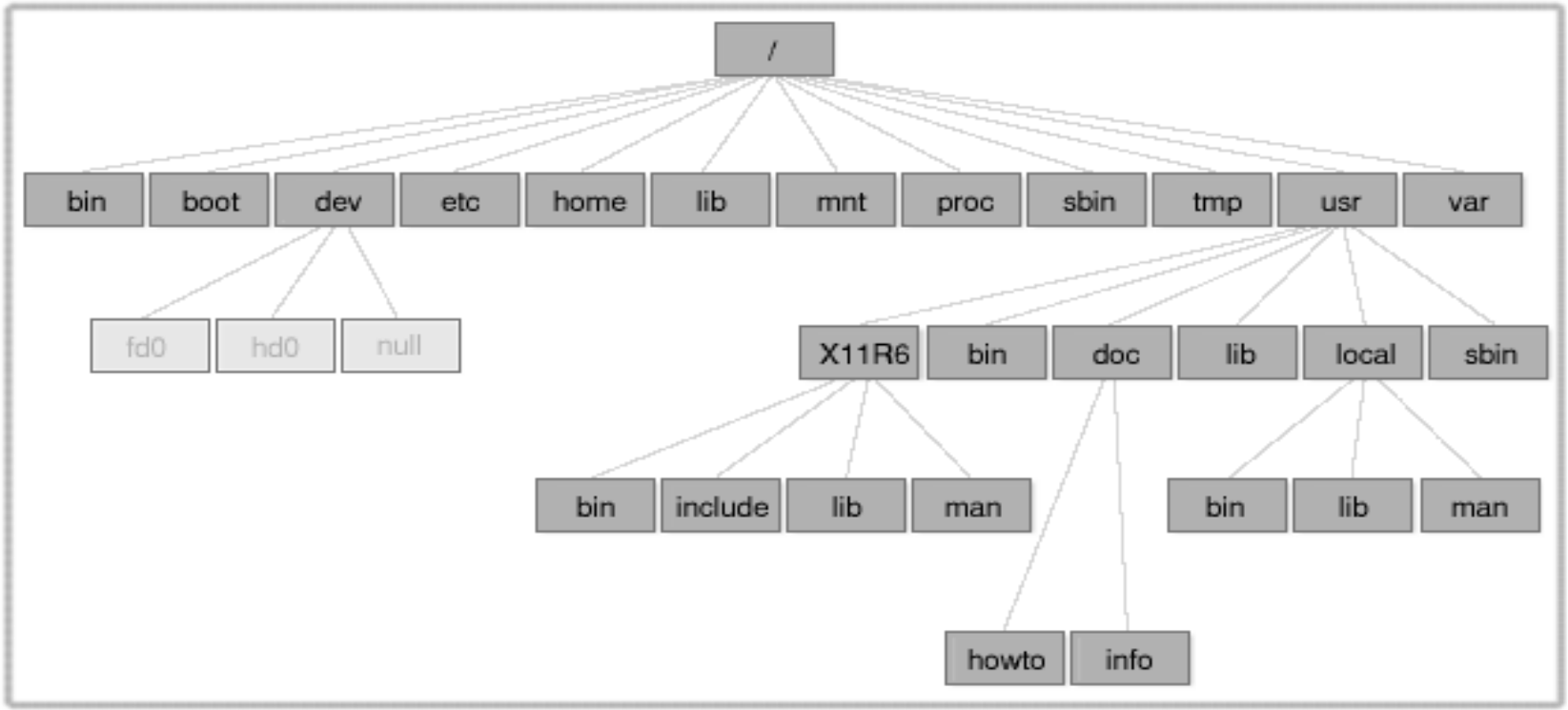
Übung

- Ausarbeiten von Übungsblatt 1 – Zeit dazu 15 Minuten
  Hinweis: Der Befehl `apropos` kann Ihnen bei der Suche nach Manpages hilfreich sein, machen Sie sich mit ihm vertraut (`man apropos`).

- Für die ganz schnellen:
  http://www2.staff.fh-vorarlberg.ac.at/~mde/linux-kurs/DOS-nach-Linux-HOWTO/DE-DOS-nach-Linux-HOWTO.html – DOS nach Linux HOWTO

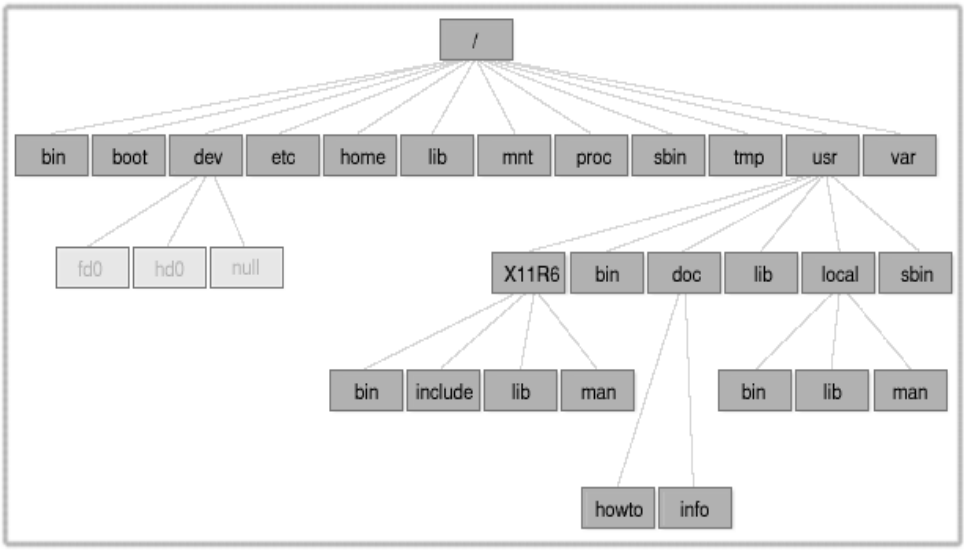


Verzeichnisstruktur



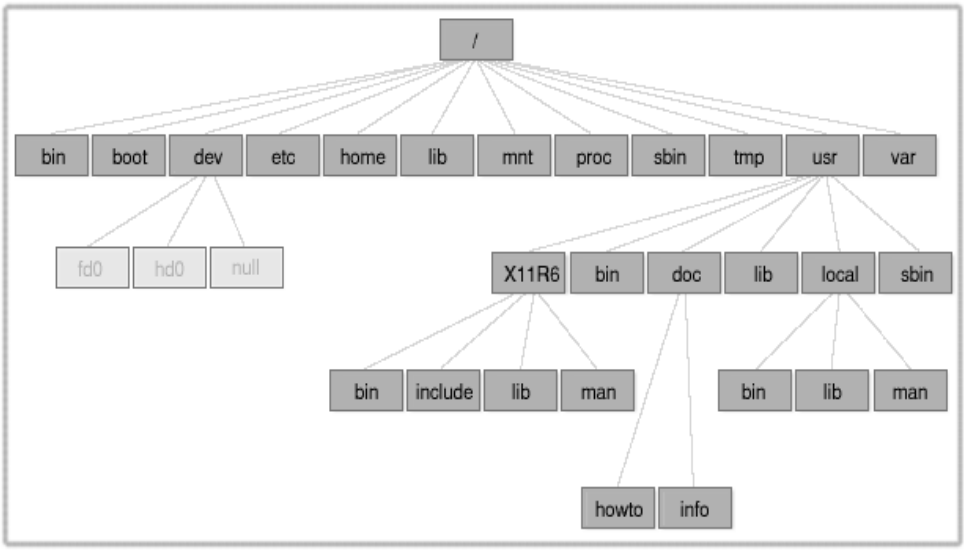
- Anfangspunkt ist die Wurzel (root)
- Jede Datei und jedes Verzeichnis ausgehend von der Wurzel erreichbar
- Entwickelt vom Filesystem Hierarchie Standard Gremium
  http://www.pathname.com/fhs/
- Liefert keinen Hinweis über Anzahl, Art der verwendeten Datenträger, Typus des/der verwendeten Filesystems/Filesysteme
- Struktur kann je nach Distribution leicht variieren (FHS Konformität)

Verzeichnisstruktur



- **/bin**
  Enthält die wichtigsten Kommandos, um mit dem System arbeiten zu können. Sie dürfen von jedem Benutzer ausgeführt werden.
- **/boot**
  Hier befinden sich die statischen Dateien des Bootmanagers, und die Kernel.
- **/dev**
  In diesem Verzeichnis stehen die Schnittstellen zur Ansteuerung der gesamten Hardware. Gemäß der Unix-Philosophie *"Alles ist eine Datei"* werden diese durch Dateien repräsentiert. In neueren Kerneln werden diese Schnittstellen dynamisch vom Kernel erzeugt und gelöscht.
- **/etc**
  Hier befinden sich die lokalen Konfigurationsdateien
- **/home**
  Alle *Heimatverzeichnisse* der Nutzer finden sich standardmässig hier. Nach dem Login landet jeder i.d.R. in seinem *Home*. (Unter Bash durch `~User` zu erreichen)
- **/lib**
  Die beim Systemstart benötigten Bibliotheken befinden sich hier. Ebenso liegen die Kernelmodule in einem eigenen Unterverzeichnis unterhalb von `/lib`.

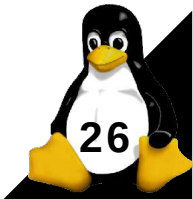
Verzeichnisstruktur



- **/mnt**
  Mountpunkt für temporäre Partitionen
- **/opt**
  Software, die nicht zum üblichen Installationsumfang von Unix-Systemen gehört, werden oft unter diesem Zweig installiert.
- **/root**
  Heimatverzeichnis des Administrators.
  In realen Unix Installationen werden die Heimatverzeichnisse der Benutzer oft auf einem zentralen Server gehalten. Bei einem Ausfall sollte sich aber zumindest root einloggen können, daher liegt dessen Heimatverzeichnis direkt unter der Verzeichnisbaumwurzel.
- **/sbin**
  Wichtige Systemprogramme (beim Booten benötigt, Ausführung erfordert root-Rechte)
- **/tmp**
  Temporäre Dateien können hier abgelegt werden, jeder Nutzer ist dazu berechtigt.
- **/usr**
  Zweite Systemhierarchie. Zur Installation “regulärer” Unix-Programme.
- **/var**
  Speicherplatz für Daten die häufiger Änderung unterliegen, oder nur für kurze Zeit existieren (zB.: zu druckende Dateien, News, Caches, ...)

Speichermedieneingliederung

- Physische Struktur bleibt dem Anwender verborgen
- Partition einer Festplatte beinhaltet dabei das Root-Dateisystem, also die Wurzel selbst.
- Alle weiteren Partitionen/Festplatten werden in die Struktur integriert, indem ein (fast) beliebiges Verzeichnis auf eine solche Partition/Festplatte verweist.
- Einbinden erfolgt durch das **mount**en (`man mount`)
  Bsp.: `mount [-t ntfs]/dev/hda3 /mnt/windows`
- Ausgliedern erfolgt durch den Befehl **umount** (`man umount`)
  Bsp.: `umount /mnt/windows`
- Standard Mount Partitionen befinden sich in `/etc/fstab`
- Dies ist gültig für jeden beliebigen Speichertyp (HDD, FDD, ..), diese unterscheiden sich nur durch ihre Dateisysteme



Auszug der wichtigsten Dateisysteme:

- **ext2:** Das Standarddateisystem von Linux
- **ext3:** Weiterentwicklung von ext2 mit zusätzlichen Journaling Funktionen
- **reiserfs:** Eines der ersten Journaling Dateisysteme für Linux
- **iso9660:** Standard-Speicherformat von Compact Discs
- **devfs:** Device File System (virtuelles Dateisystem zur Verwaltung der Geräte) -> `/dev`
- **loopback:** Mounten einer einzelnen Datei als Dateisysten (zB.: CD-Images)
- **nfs:** Standard Unix Netzwerk-Dateisystem
- **ntfs:** Windows 2000 / XP Dateisystem, Schreibender Zugriff nur beschränkt möglich
- **tmpfs:** Mounten eines Bereichs des Hauptspeichers als Dateisystem
- **smbfs:** (Server Message Block) Windows Netzwerk-Dateisystem
- **swap:** Swap Partitionen oder Dateien

- `cat /proc/filesystems` zeigt die aktuell vom Kernel unterstützen Dateisysteme an
- Beispielsausgabe des `mount` Befehles:

```
/dev/hda3 on / type reiserfs (rw,noatime,commit=360)
none on /dev type devfs (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw)
none on /dev/shm type tmpfs (rw)
none on /tmp type tmpfs (rw,size=128m,mode=1777)
none on /proc/bus/usb type usbfs (rw)
192.168.0.254:/home/pool on /home/marco/pool type nfs (rw,addr=192.168.0.254)
```

Dateitypen & Rechte

- Dateitypen werden unabhängig vom Dateisystem betrachtet, dass Dateisystem kann nur Parameter des Typs beeinflussen

- **Dateitypen:**

Symbol	Bedeutung
-	Normale Datei
b	Blockorientiertes Gerät (zB.: Harddisk)
c	Zeichenorientiertes Gerät (zB.: Terminal)
d	Verzeichnis
l	Symbolischer Link
p	FIFO-Datei (named Pipe)
s	Unix Domain Socket

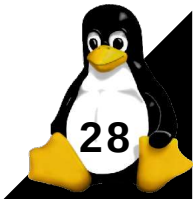
Dateirechte:

Drei Gruppen zu je drei Spalten für die Rechte
-**rwx**rwxrwx des Eigentümers
-rwx**rwx**rwx der Gruppe
-rwxrwx**rwx** der anderen

r	darf gelesen werden (read)
w	darf geschrieben werden (write)
x	darf ausgeführt werden (execute)
s	Setuid Bit

- root befindet sich ausserhalb dieser Dateirechte Struktur, er darf daher alles
- Beispiele:

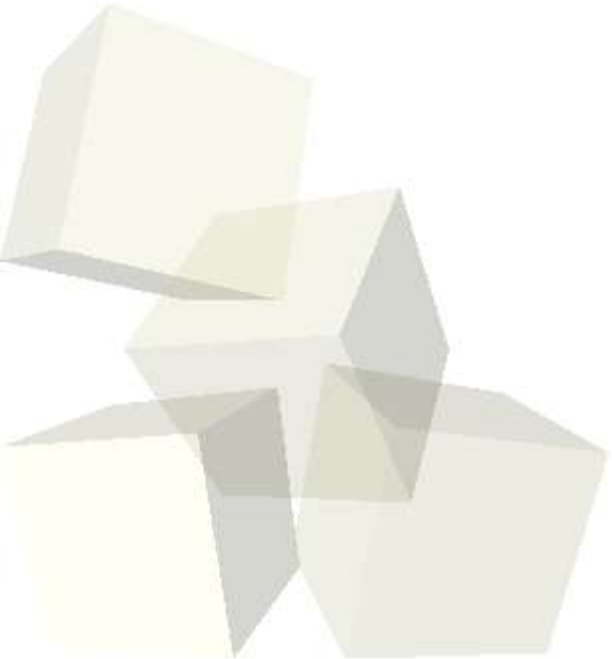
```
crw-rw-rw-  1 root root 1, 5 Jan  1  1970 /dev/zero
drwxr-xr-x  4 marco users  240 Aug 16 09:35 linux-kurs
-rw-r--r--  1 marco users 107 Aug 18 09:44 dateirechte.txt
-rwxr-xr-x  1 root root 596K Jul 15 19:17 /bin/bash
lrwxrwxrwx  1 root root 10 Aug  2 13:22 /lib/libss.so -> libss.so.2
```



Übung

- Ausarbeiten von Übungsblatt 2 + 3 – Zeit dazu 30 Minuten

- Für die ganz schnellen:
  - Machen Sie sich mit der Datei `/etc/fstab` vertraut, was ist der Sinn dieser Datei bzw. interpretieren Sie die darin stehenden Daten.
  - Der `stat` Befehl liefert Ihnen interessante Zusatzinformationen über eine Datei oder ein Filesystem, machen Sie sich mit Ihm vertraut, und interpretieren Sie die Ausgabe.

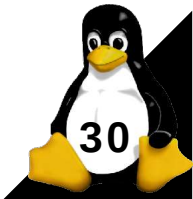


Anzeigen und Editieren

- Befehle zum Anzeigen von Dateien:
  - `cat` dient zum Anzeigen von Dateien, kann aber auch zum Anhängen einzelner Zeilen verwendet werden.
    Bsp.: Anzeigen von Datei: `cat dateiname`
    Anhängen an Datei : `cat > dateiname` beenden mit `STRG+D`
  - `less, more` zeigen Datei jeweils seitenweise an
    Bsp.: `less dateiname` beenden mit `q`

- Editoren unter Linux:
  - `vi(m)` der Unix Editor, auf jedem *nix System zu finden, relativ komplex in der Anwendung http://www.vim.org
    - **Überlebenshilfe:** Bei unfreiwilligem `vi` start, Eingabe von `ESC :q!` beendet `vi`. :)
  - `nano` benutzerfreundlicher Editor, pico Clone
  - `joe` freier ASCII Editor für Unix

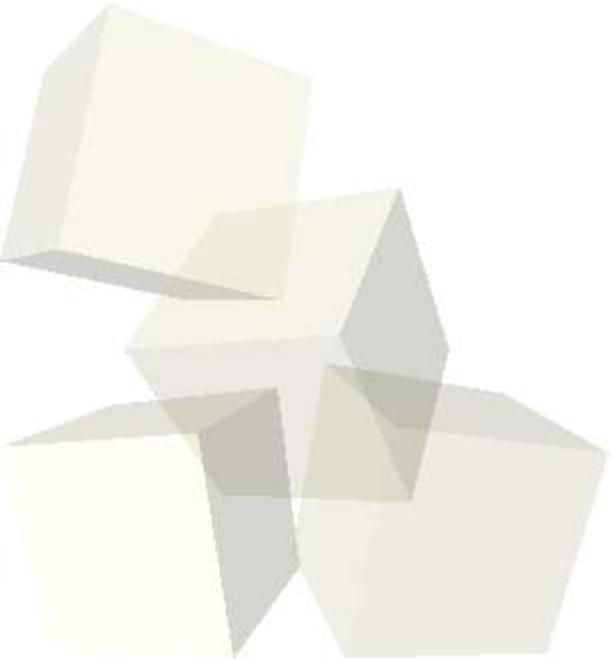
- Graphische Texteditoren bzw. -editoren für KDE usw. existieren natürlich zur Genüge (zB.: gvim, xemacs, ...)

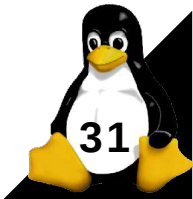


Übung

- Ausarbeiten von Übungsblatt 4 – Zeit dazu 10 Minuten

- Für die ganz schnellen:
  - Arbeiten Sie sich in den Standard-Editor von Unix Systemen `vi` ein, sie werden ihn auf nahezu jedem Unix-System finden, und er funktioniert auch über langsame Remote-Verbindungen gut, hier einige Links zur Hilfe:
    http://www.fh-wedel.de/~di/html/vi/ - VI-Kurzreferenz
    http://www.napali.ch/Einfuehrung_in_Vim/ - Einführung in Vim
    http://www.smial.de/vi.html - Kurze Einführung in den Editor vi





Bash Konzepte

- Metazeichen bzw. Wildcards

`?`	Genau ein beliebiges Zeichen	`*`	Beliebig viele (auch 0) Zeichen
`[ab]`	Eines der beiden Zeichen	`[!ab]`	Keines der beiden Zeichen
`~`	Heimatverzeichnis	`\`	Escape (zB.: \" um " zu meinen)

  Beispiele: `ls /boot/*map` erzeugt Ausgabe `/boot/System.map`

- Eingabehilfen
  - Datei- und Kommandonamen Expansion zB.: `ls ~ro[TAB]`
  - Bash History

    Eingabe von `history` zeigt Liste aller bisher verwendeten Befehle mit ID Befehle kann durch Eingabe von `!id` wiederholt werden. (Cursortasten spulen in History)
  - Alias Unterstützung, Eingabe von `alias` zeigt momentane Definitionen

- Eingabe/Ausgabe Umleitung

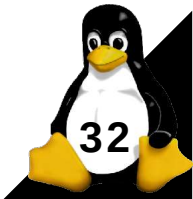
`< datei`	Standardeingabe lesen aus `datei`
`> datei`	Standardausgabe umleiten in `datei`
`>> datei`	Standardausgabe anhängen an `datei`

- Pipes

  Dienen der Verknüpfung eines Ausgabe mit einem Eingabestrom

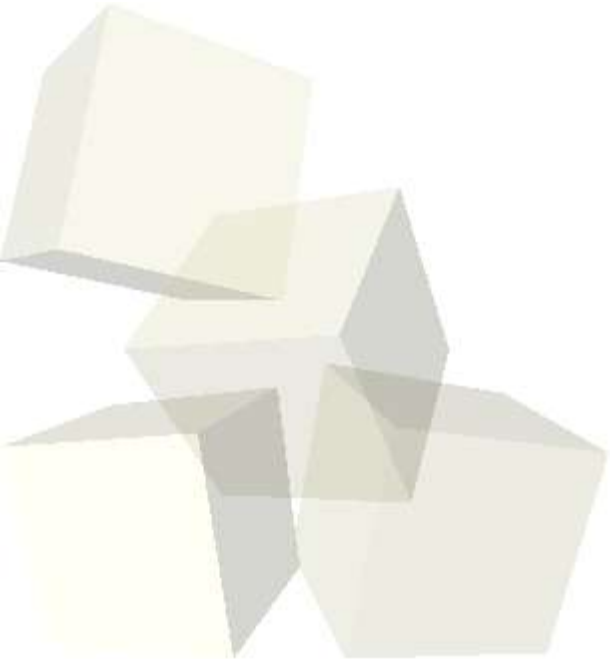
  Beispiele: `ls -l | less` zeigt Ausgabe seitenweise an

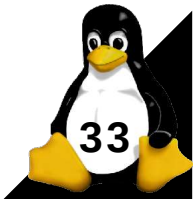
  `cat test.txt | grep hallo` zeigt Zeilen mit hallo in `test.txt`



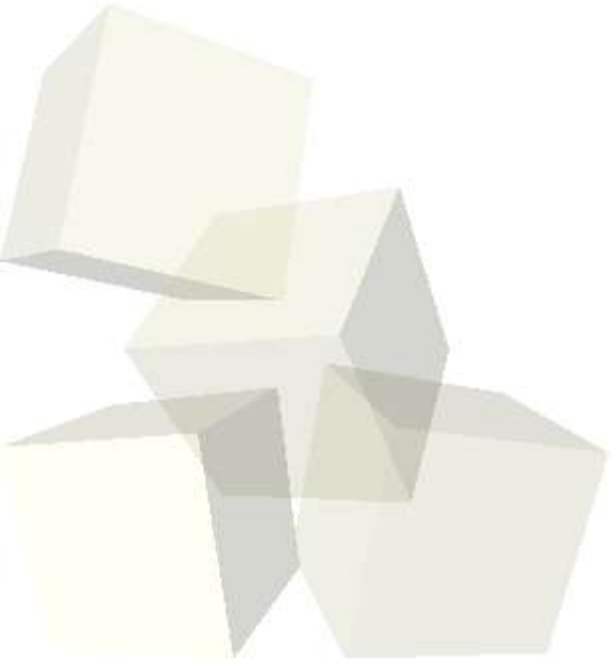
Übung

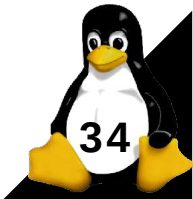
- Ausarbeiten von Übungsblatt 5 – Zeit dazu 10 Minuten
- Für die ganz schnellen:
  - http://www.selflinux.org/selflinux/html/bash_basic.html - Bash Einführung





Prozess- und Logmanagement





Prozessmanagement

- Prozesse bilden tragendes Konzept jedes Betriebssytems
- Prozesse werden gestartet, angehalten, reaktiviert, beendet, ...
- Maßnahmen die Prozess beeinflussen → Prozessmanagement

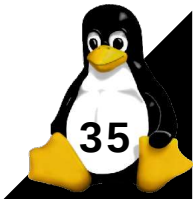
- Programme sind Ablaufpläne, Prozesse die Instanzen davon

- Prozesse befinden sich in einer Verwandschaft miteinander
  - Ursprung aller Prozesse ist `init` er besitzt immer die Prozessidentität (PID) 1, er startet alle weiteren Programme (childs, parents, orphants, zombies,..)

- Definition Prozess:
  Ein Stück Programm-Code, das zur Ausführung in den Hauptspeicher geladen wurde, und im System durch Parameter wie Prozessnummer (PID), Elternprozessnummer (PPID), Besitzer, Priorität, Nice-Level, Status usw. gekennzeichnet ist.

- Anzeige der aktuellen Prozesse des Users mittels `ps`
  Beispielausgabe `ps` einer Konsole:

```
  PID TTY          TIME CMD
 9074 pts/5    00:00:00 bash
18406 pts/5    00:00:00 ps
```

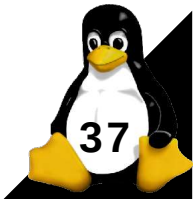


Beispielausgabe durch `pstree -p`

```
init(1)-+-acpid(5840)
        |-agetty(6185)
        |-bonobo-activati(6748)
        |-cron(6127)
        |-devfsd(419)
        |-dhcpcd(6437)
        |-events/0(3)-+-aio/0(48)
        |             |-kacpid(5)
        |             |-kblockd/0(27)
        |             |-khelper(4)
        |             |-pdflush(45)
        |             |-pdflush(46)
        |             `-reiserfs/0(202)
        |-gconfd-2(6540)
        |-gdm(6257)---gdm(6259)-+-X(6270)
        |                       `-fluxbox(6394)-+-firefox(17458)---firefox-bin(17465)
        |                                       |
        |                                       |-gkrellm2(6413)---gkrellm2(6431)
        |                                       |-soffice.bin(6770)---soffice.bin(6800)-+-soffice.bin(6801)
        |                                       |                                       |-soffice.bin(6802)
        |                                       |                                       `-soffice.bin(6825)
        |                                       |-ssh-agent(6412)
        |                                       |-xterm(6477)---bash(6479)---micq(6484)
        |                                       |-xterm(6486)---bash(6488)
        |                                       `-xterm(7122)---bash(7124)---pstree(18302)
        |-heartbeat-clien(6470)
        |-khubd(28)
        |-kseriod(51)
        |-ksoftirqd/0(2)
        |-kswapd0(47)
        |-metalog(5799)---metalog(5800)
        |-pccardd(151)
        |-smbiod(6445)
        |-smbmount(6441)
        `-speedfreqd(6087)
```

Prozessmanagement

- Ein Prozess kann sich in drei Zuständen befinden
  - *Rechnend* – Prozess wird gerade vom Prozessor abgearbeitet
  - *Rechenbereit* – Prozess ist ausführbar, wartet aber auf Rechenzeitzuteilung
  - *Blockiert* – Prozess wartet auf ein externes Ereignis
- Prozessarten
  - *Interaktive Prozesse* – normales Programm mit Ein-/Ausgabe durch Nutzer
  - *Daemons* – Serverprozess der im Hintergrund seine Arbeit verrichtet
  - *Batch Prozesse* – Nicht an Terminal oder Session gebunder Prozess
- Prozessmanagement Befehle
  - `ps` – Prozesstatus anzeigen
  - `top` – Linux Tasks anzeigen (erlaubt einfach Steuerung)
  - `kill` – Signal an einen Prozess senden (zB.: Zur Steuerung von Daemons)
  - `killall` – Prozess ein Signal senden durch Namensidentifikation
  - `nice` – Prozess mit modifizierter Priorität starten
  - `renice` – Priorität eines laufenden Prozesses verändern



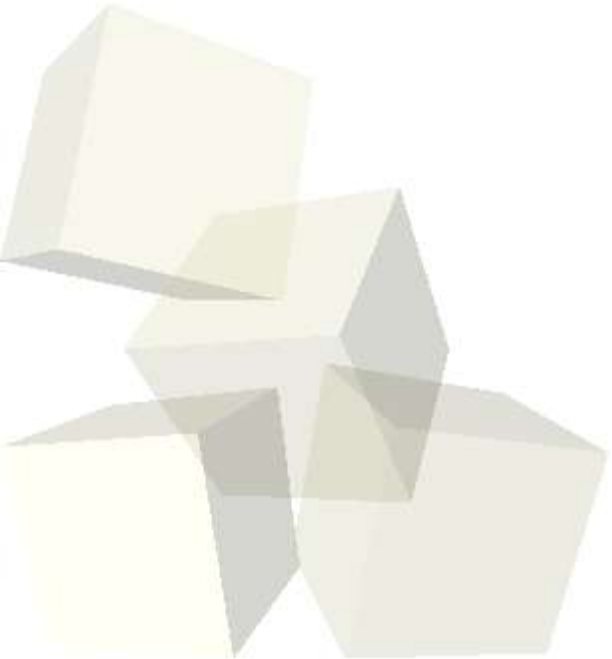
Beispielausgabe des `top` Befehls:

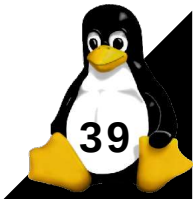
```
top - 14:41:05 up  4:11,  6 users,  load average: 0.00, 0.06, 0.07
Tasks:  68 total,   1 running,  67 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3% us,  0.3% sy,  0.0% ni, 99.3% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:    905564k total,   879644k used,    25920k free,   242324k buffers
Swap:  1004052k total,        0k used,  1004052k free,   316776k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
20148 marco     16   0  1852  996 1644 R  0.3  0.1   0:00.01 top
    1 root      16   0  1308  476 1156 S  0.0  0.1   0:01.02 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      10 -10     0    0    0 S  0.0  0.0   0:00.00 khelper
    5 root       5 -10     0    0    0 S  0.0  0.0   0:05.77 kacpid
   27 root       5 -10     0    0    0 S  0.0  0.0   0:00.27 kblockd/0
   28 root      15   0     0    0    0 S  0.0  0.0   0:00.21 khubd
   45 root      15   0     0    0    0 S  0.0  0.0   0:00.08 pdflush
   46 root      15   0     0    0    0 S  0.0  0.0   0:00.03 pdflush
   48 root       7 -10     0    0    0 S  0.0  0.0   0:00.00 aio/0
   47 root      15   0     0    0    0 S  0.0  0.0   0:00.04 kswapd0
   51 root      17   0     0    0    0 S  0.0  0.0   0:00.00 kseriod
  151 root      15   0     0    0    0 S  0.0  0.0   0:00.00 pccardd
  195 root       5 -10     0    0    0 S  0.0  0.0   0:00.00 reiserfs/0
  412 root      18   0  1660  932 1348 S  0.0  0.1   0:00.01 devfsd
 5792 root      16   0  1896  596 1212 S  0.0  0.1   0:00.03 metalog
 5793 root      16   0  1368  484 1212 S  0.0  0.1   0:00.00 metalog
 5833 root      16   0  1440  548 1144 S  0.0  0.1   0:00.04 acpid
 6080 root      16   0  1304  484 1156 S  0.0  0.1   0:00.00 speedfreqd
 6120 root      16   0  1616  640 1324 S  0.0  0.1   0:00.00 cron
 6178 root      17   0  1436  644 1256 S  0.0  0.1   0:00.02 agetty
 6179 root      16   0  1436  644 1256 S  0.0  0.1   0:00.00 agetty
 6180 root      17   0  1436  644 1256 S  0.0  0.1   0:00.00 agetty
 6181 root      17   0  1436  644 1256 S  0.0  0.1   0:00.00 agetty
 6182 root      17   0  1436  644 1256 S  0.0  0.1   0:00.00 agetty
 6183 root      17   0  1436  644 1256 S  0.0  0.1   0:00.00 agetty
 6250 root      16   0  8252 2216 8044 S  0.0  0.2   0:00.00 gdm
 6252 root      16   0  8768 2848 8316 S  0.0  0.3   0:00.04 gdm
 6263 root      15   0 88064  72m  17m S  0.0  8.2   3:40.03 X
 6387 marco     16   0  5680 3200 4948 S  0.0  0.4   0:02.38 fluxbox
```

Übung

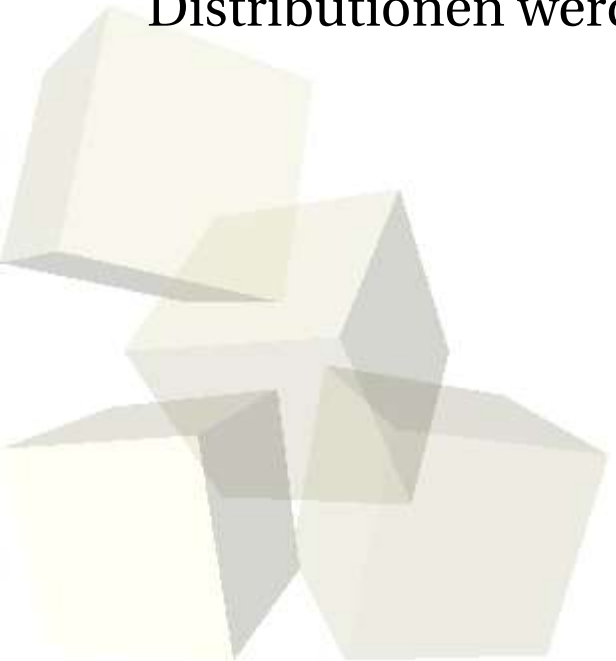
- Ausarbeiten von Übungsblatt 6 – Zeit dazu 15 Minuten
- Für die ganz schnellen:
  - http://www.tu-chemnitz.de/urz/kurse/unterlagen/linux-bedienung/KDE3/sonstiges/prozessmgm.htm - Prozessmanagement

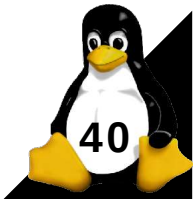




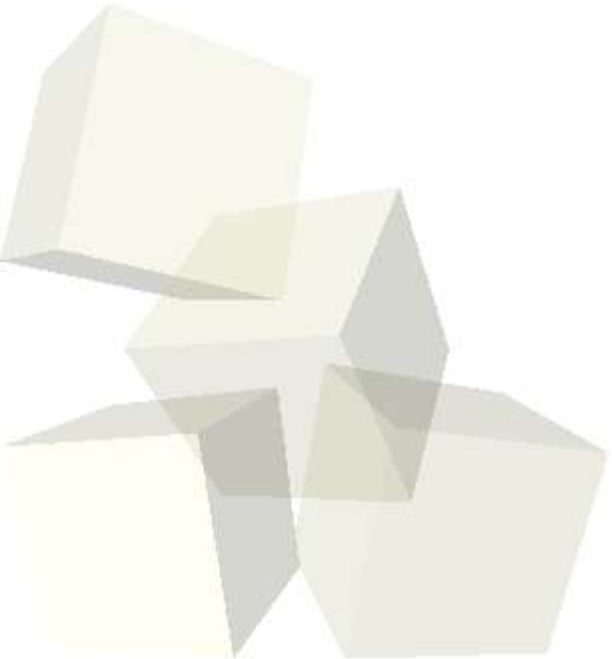
Logmanagement

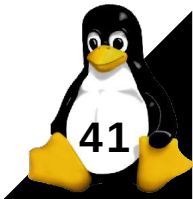
- Logfiles sind Dateien die Nachrichten über das System, den Kernel, die Dienste und die laufenden Applikationen enthalten.
- Normalerweise existieren auf *nix Systemen mehrere Logfiles, welche sich bestimmten Diensten bzw. Bereichen widmen (zB.: Security, Apache-Logfiles)
- Die meisten Logfiles befinden sich im `/var/log` Verzeichnis und sind im plaintext-Format geschrieben, was bedeutet dass ein Standardanzeigebefehl verwendet werden kann
- Der `dmesg` Befehl zeigt die letzten Ausgaben des Kernels an, in machen Distributionen werden Kernelnachrichten direkt auf den Bildschirm angezeigt





Anwendermanagement





Anwendermanagement

- Die Daten zur Benutzerverwaltung findet man in `/etc/passwd`, die zur Gruppenverwaltung in `/etc/group`

- Der Begriff Benutzer ist etwas weit gefasst, da auch Pseudobenutzer existieren, um Rechte für bestimmte Dateien/Verzeichnisse/Prozesse nur bestimmten Programmen einzuräumen.

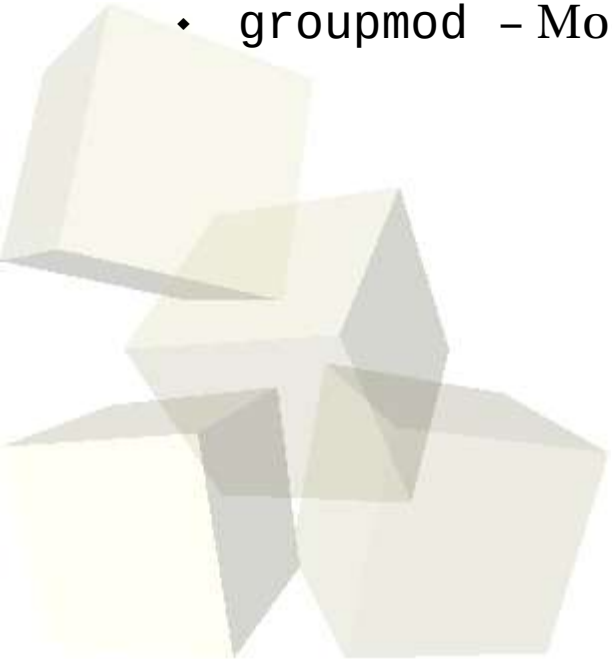
- Der Aufbau innerhalb von `/etc/passwd` ist immer gleich:
  `Username : Password : UID : GID : Info : Home : Shell`

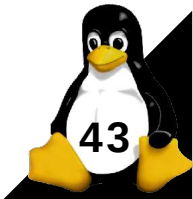
- Die Passwörter an sich befinden sich nur bei älteren Systemen im `passwd` File, bei neueren Systemen befindet sich im Password Feld nur ein X, welches darauf hinweist dass das eigentliche unter `/etc/shadow` zu finden ist.

Anwendermanagement

- Folgende Befehler stehen zur Benutzerverwaltung zur Verfügung:
  - `useradd` – Fügt einen neuen User hinzu
  - `userdel` – Löscht einen bestehenden User
  - `usermod` – Modifiziert die Daten eines bestehenden Benutzers
  - `chfn` – Username und Informationen ändern
  - `chsh` – Standard Shell des Users ändern
  - `chage` – Ablaufzeit des Passworts eines Users ändern
  - `passwd` – Passwort eines Users ändern

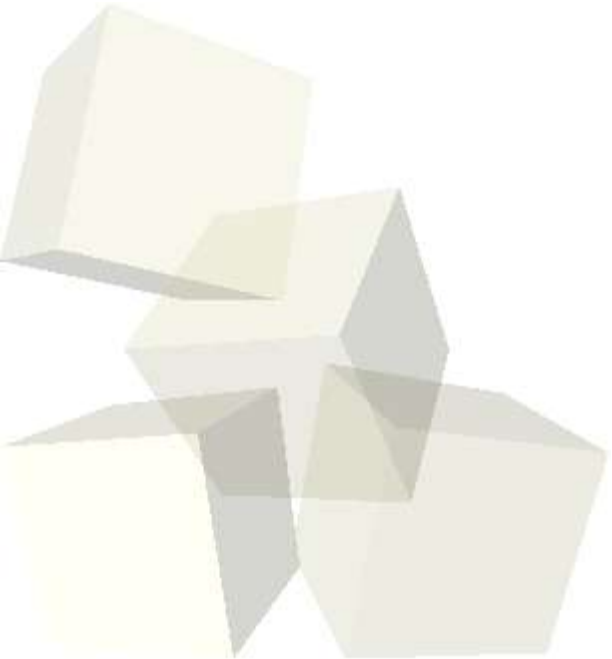
- Folgende Befehle stehen zur Gruppenverwaltung zur Verfügung:
  - `groupadd` – Fügt eine neue Gruppe zum System hinzu
  - `groupdel` – Löscht eine bestehende Gruppe
  - `groupmod` – Modifiziert eine bestehende Gruppe

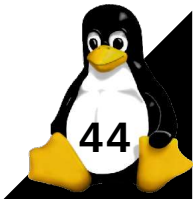




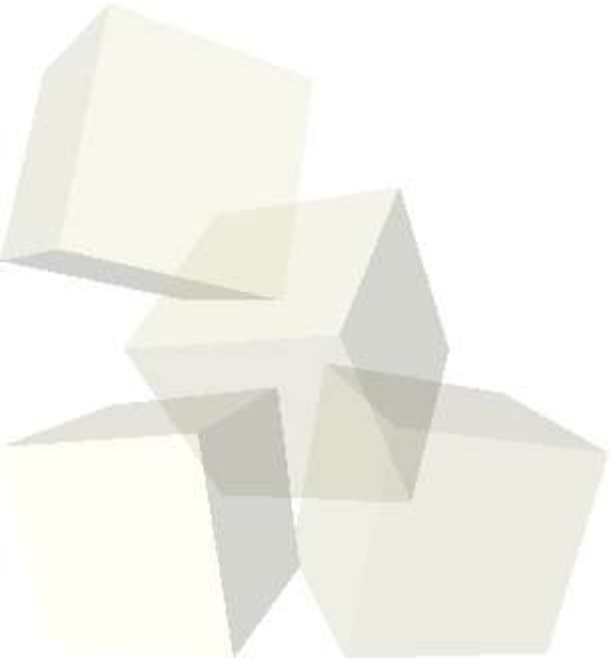
Übung

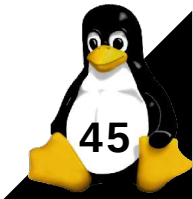
- Ausarbeiten von Übungsblatt 7 – Zeit dazu 15 Minuten
- Für die ganz schnellen:
  - http://www.linuxfibel.de/useradmin.htm - Benutzerverwaltung





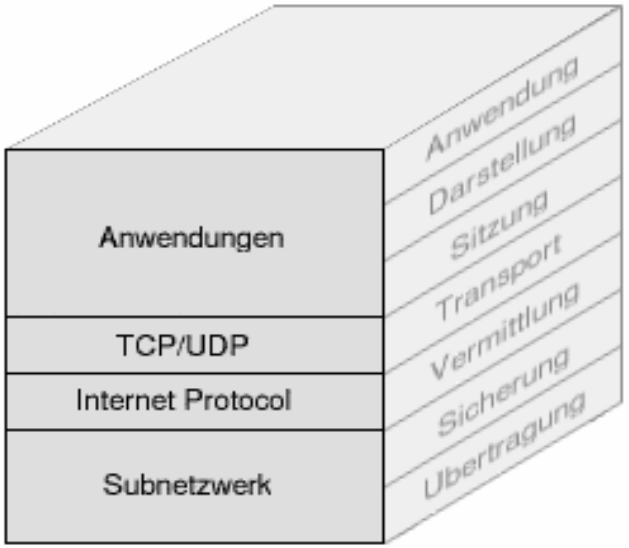
Grundkonzepte Netzwerk





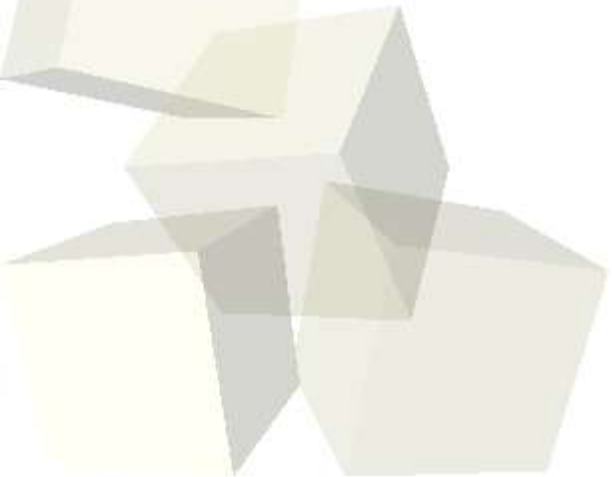
Kurzeinführung IP

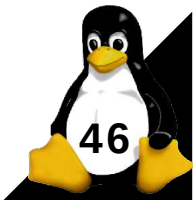
- **IP(v4)** stellt Basis des Internet Datenverkehrs und ermöglicht mittels **TCP, UDP** und **ICMP** die Kommunikation
  - **TCP/IP –** verbindungsorientierte "Verbindungen"
  - **UDP/IP –** Datagramme (verbindungslos)
  - **ICMP** – Kontrolldaten zur reibungslosen Komm.



- **IP** addressiert anhand 32-bittiger Adressen, welche in Netzwerkklassen eingeteilt werden.
  Bsp.: `192.168.0.254/255.255.255.0`

- Detailierte Informationen über IP findet man problemlos im Internet zB unter:
  http://www.physio.uni-luebeck.de/schulung/_private/netzwerk/tcpip.html





Netzwerkkonfiguration

- Grundlegende Tools zur Netzwerkkonfiguration unter Linux:
  - `ifconfig` – Konfiguriert ein Netzwerkinterface

```
eth0      Link encap:Ethernet  HWaddr 00:C0:9F:3A:84:17
          inet addr:193.170.2.249  Bcast:193.170.2.255  Mask:255.255.255.224
          inet6 addr: fe80::2c0:9fff:fe3a:8417/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:77844 errors:0 dropped:0 overruns:0 frame:0
          TX packets:148555 errors:0 dropped:0 overruns:0 carrier:5
          collisions:0 txqueuelen:1000
          RX bytes:68650874 (65.4 Mb)  TX bytes:13557354 (12.9 Mb)
          Interrupt:10
```

  - `route` – Zeigt bzw. manipuliert die Routingtabelle

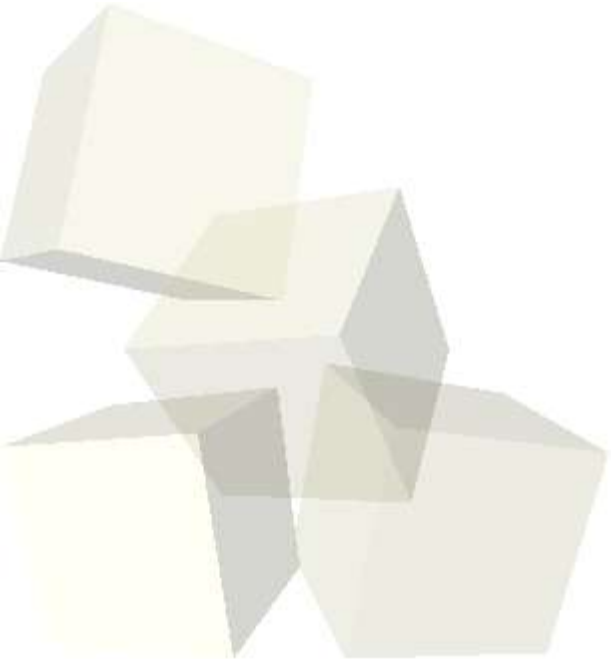
```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
193.170.2.224   *               255.255.255.224 U     0      0        0 eth0
loopback        localhost       255.0.0.0       UG    0      0        0 lo
default         router.p.uclv.n 0.0.0.0         UG    0      0        0 eth0
```

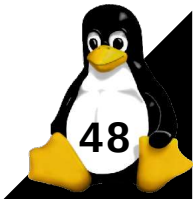
  - `netstat` – Zeigt allgemeine Netzwerkdaten
  - `arp` – Zur Bearbeitung des System ARP-Caches

```
Address                  HWtype  HWaddress           Flags Mask            Iface
router.p.uclv.net        ether   00:E0:52:AD:1F:00   C                     eth0
```

Übung

- Ausarbeiten von Übungsblatt 8 – Zeit dazu 20 Minuten
- Für die ganz schnellen:
  http://www.linux-related.de/index.html?/admin/netzwerk.htm - Manuelle Netzwerk-konfiguration unter Linux



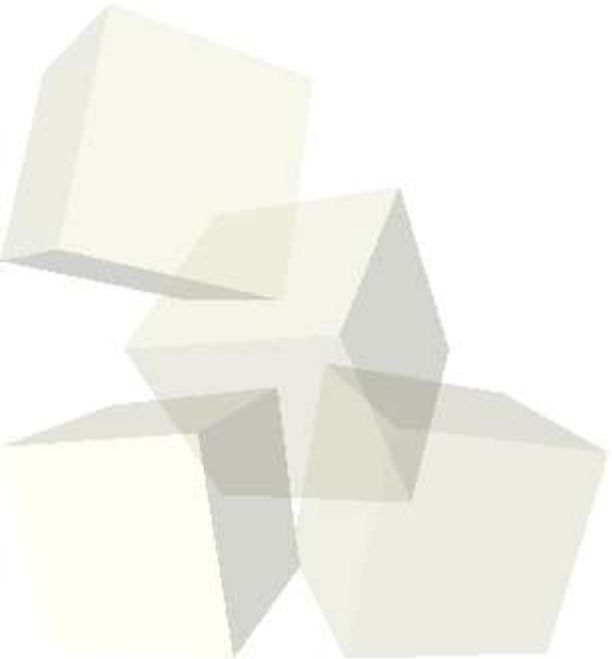


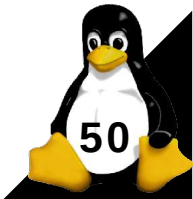
Remote Access

- Folgende Methoden zur entfernten Administration stehen unter Linux zur Verfügung (Auswahl unvollständig):
  - **telnet** unverschlüsselte remote shell
  - **openssh** verschlüsselte remote shell, ermöglicht auch XFree-Weiterleitung (www.openssh.org)
  - **webmin** web basierte remote administration (www.webmin.com)
- **openssh** stellt dabei die gängigste und wichtigste Methode für einen Administrator dar

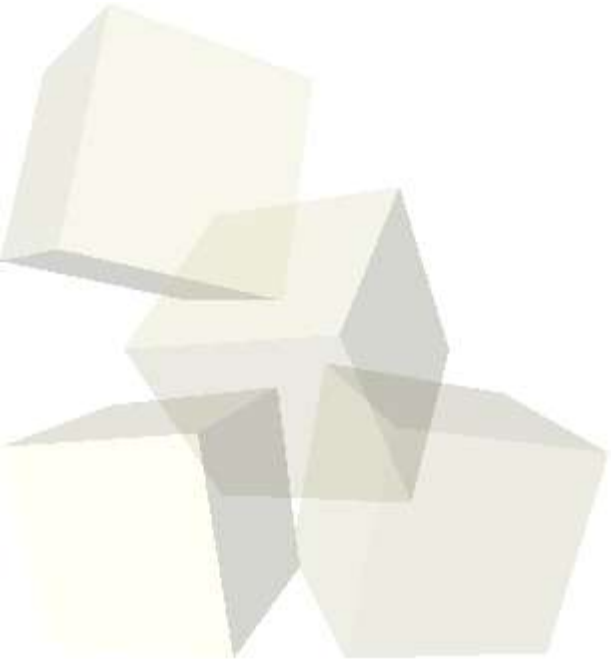
Übung

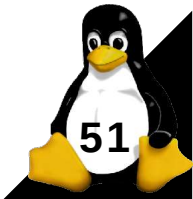
- Ausarbeiten von Übungsblatt 9 – Zeit dazu 10 Minuten
- Für die ganz schnellen:
  - http://www.openssh.org/faq.html - OpenSSH Project FAQ





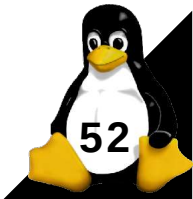
Netzwerkdateisysteme





Netzwerkdateisysteme

- **NFS** – Network File System
  - Standard Netzwerk Dateisystem für Linux (und andere Unices)
  - Existiert mittlerweile experimentell in V4 (unterstützt Verschlüsselung)
  - http://www.nfsv4.org
- **SMBFS** – Samba File System
  - Ermöglicht Exporte an Windows Computer
  - In neuen Versionen kann Samba Domain Controller Funktionalitäten übernehmen
  - http://www.samba.org
- **NETATALK** – Apple Protocol Suite Implementation
  - Ermöglicht Exporte an Macintosh Computer
  - Bei MacOsX nicht mehr notwendig, da selbst Unix basiert
  - http://netatalk.sourceforge.net
- Alternativen werden durch verteilte Dateisysteme geboten
  - Beispiele: Coda, Andrew File System, Global File System



NFS Konfiguration

- Zum Betrieb eines NFS-Servers sind zwei Komponenten notwendig:
  - `nfsd` – Der NFS Daemon (Server) an sich
  - `portmap` – Der Portmap Server für die Socket Zuteilung

- Die Konfiguration der zu exportierenden Verzeichnisse kann über zwei Wege erfolgen:
  - `/etc/exports` – Die statische Konfigurationsdatei für Exporte

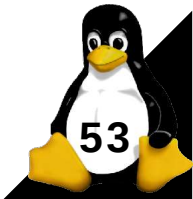
```
#   $OpenBSD: exports,v 1.2 2002/05/31 08:15:44 pjanzen Exp $
#
# NFS exports Database
# See exports(5) for more information.  Be very careful:  misconfiguration
# of this file can result in your filesystems being readable by the world.
/home -alldirs -network 192.168.0 -mask 255.255.255.0
```

  - `exportfs` – Befehl zum dynamischen Export

- Das Einbinden eines NFS-Exports auf einem Client erfolgt mittels des Standard `mount` Befehls, durch Angabe des Parameters `-t nfs`

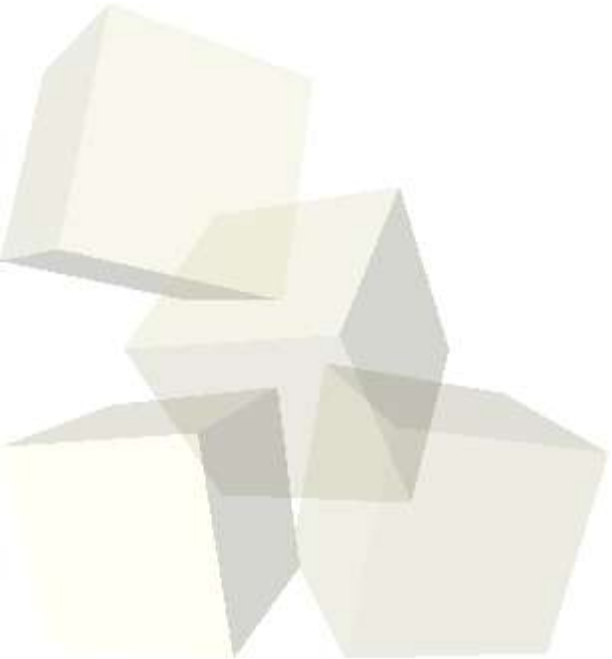
```
mount -t nfs 192.168.0.254:/home/pool /home/marco/home-shares/pool
mount -t nfs 192.168.0.254:/home/marco /home/marco/home-shares/marco
```

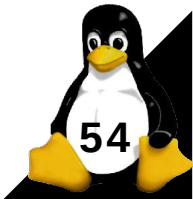
Weitere Infos: http://www.linuxfibel.de/nfs_srv.htm bzw. http://www.linuxfibel.de/nfs_cli.htm



Übung

- Ausarbeiten von Übungsblatt 10 – Zeit dazu 20 Minuten
- Für die ganz schnellen:
  http://www.linuxbu.ch/pdf/155_164.pdf - NFS einrichten





Systemsicherheit