Pinguin Kindergarten

Pinguin Kindergarten Slides von Stefan Walluhn steht unter einer
Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz.

Über ...

About me

(jabber+mail: stefan@terminal21.de)

*

Terminal.21

*

Basislager

Erwartungen?

Closed Source

```
00000000  7f 45 4c 46 01 01 01 00  00 00 00 00 00 00 00 00
00000010  02 00 03 00 01 00 00 00  54 80 04 08 34 00 00 00
00000020  00 00 00 00 00 00 00 00  34 00 20 00 01 00 00 00
00000030  00 00 00 00 01 00 00 00  00 00 00 00 00 80 04 08
00000040  00 80 04 08 74 00 00 00  74 00 00 00 05 00 00 00
00000050  00 10 00 00 b0 04 31 db  43 b9 69 80 04 08 31 d2
00000060  b2 0b cd 80 31 c0 40 cd  80 48 65 6c 6c 6f 20 77
00000070  6f 72 6c 64
00000074
```

(Quelle: http://stackoverflow.com/questions/284797/hello-world-in-less-than-20-bytes)

Open Source

```
#include <stdio.h>

int main(void)
{
  printf("Hello, World\n");

  return 0;
}
```

Free Software – 4 Freiheiten

- Die Freiheit, das Programm für jeden Zweck zu benutzen (Freiheit 0).
- Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann (Freiheit 1). Der Zugang zum Quellcode ist dafür Voraussetzung.
- Die Freiheit, Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann (Freiheit 2).
- Die Freiheit, das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann (Freiheit 3). Der Zugang zum Quellcode ist dafür Voraussetzung.

(Quelle: http://www.gnu.org/philosophy/free-sw.de.html)

Copyleft – Copyright Hack

CC BY-SA 3.0

Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.

(Quelle: http://creativecommons.org/licenses/by-sa/3.0/de/)

GPLv3

5. Übertragung modifizierter Quelltextversionen

Sie dürfen ein auf dem Programm basierendes Werk oder die nötigen Modifikationen, um es aus dem Programm zu generieren, kopieren und übertragen in Form von Quelltext unter den Bestimmungen von §4, vorausgesetzt, daß Sie zusätzlich alle im folgenden genannten Bedingungen erfüllen:

> […]
> Das veränderte Werk muß auffällige Vermerke tragen, die besagen, daß es unter dieser Lizenz einschließlich der gemäß §7 hinzugefügten Bedingungen herausgegeben wird. Diese Anforderung wandelt die Anforderung aus §4 ab, „alle Hinweise intakt zu lassen".
> Sie müssen das Gesamtwerk als Ganzes gemäß dieser Lizenz an jeden lizensieren, der in den Besitz einer Kopie gelangt. Diese Lizenz wird daher – ggf. einschließlich zusätzlicher Bedingungen gemäß §7 – für das Werk als Ganzes und alle seine Teile gelten, unabhängig davon, wie diese zusammengepackt werden. Diese Lizenz erteilt keine Erlaubnis, das Werk in irgendeiner anderen Weise zu lizensieren, setzt aber eine derartige Erlaubnis nicht außer Kraft, wenn Sie sie diese gesondert erhalten haben.

(Quelle: http://www.gnu.de/documents/gpl.de.html)

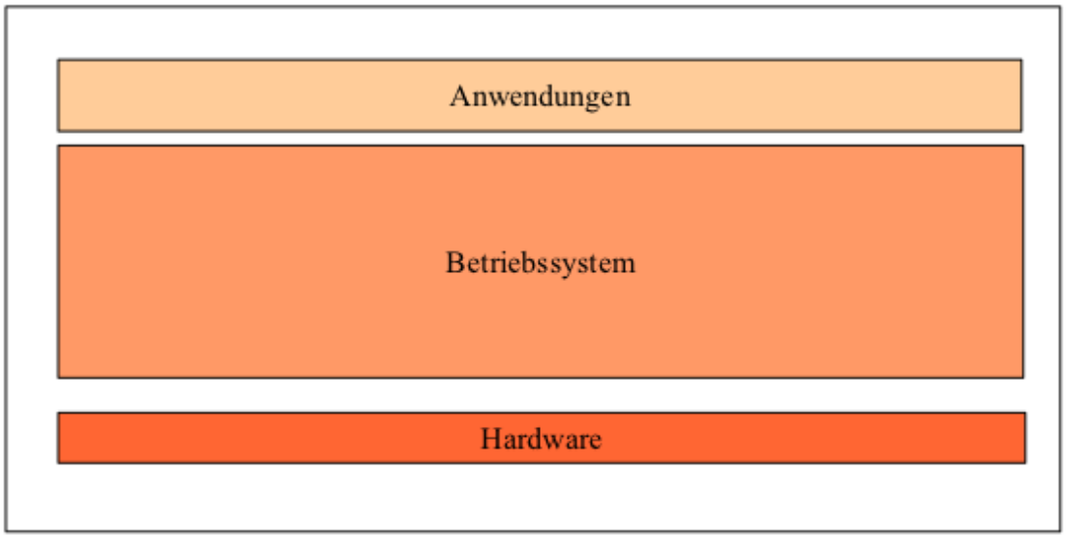
Bemerkungen

- OpenSource != Free Software
- OpenSource != Copyleft
  - Bsp: GPL vs. BSD-License (welche Lizenz ist die „freiere“ Lizenz?)
- Free Software != kostenlos
  - “Free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”.
- Bsp. OpenSource: Linux, BSD, Darwin
- Bsp. FreeSoftware: Linux, Android, GNU
- Bsp. ClosedSource (proprietär): Windows, MS-DOS, SymbianOS

Betriebssysteme

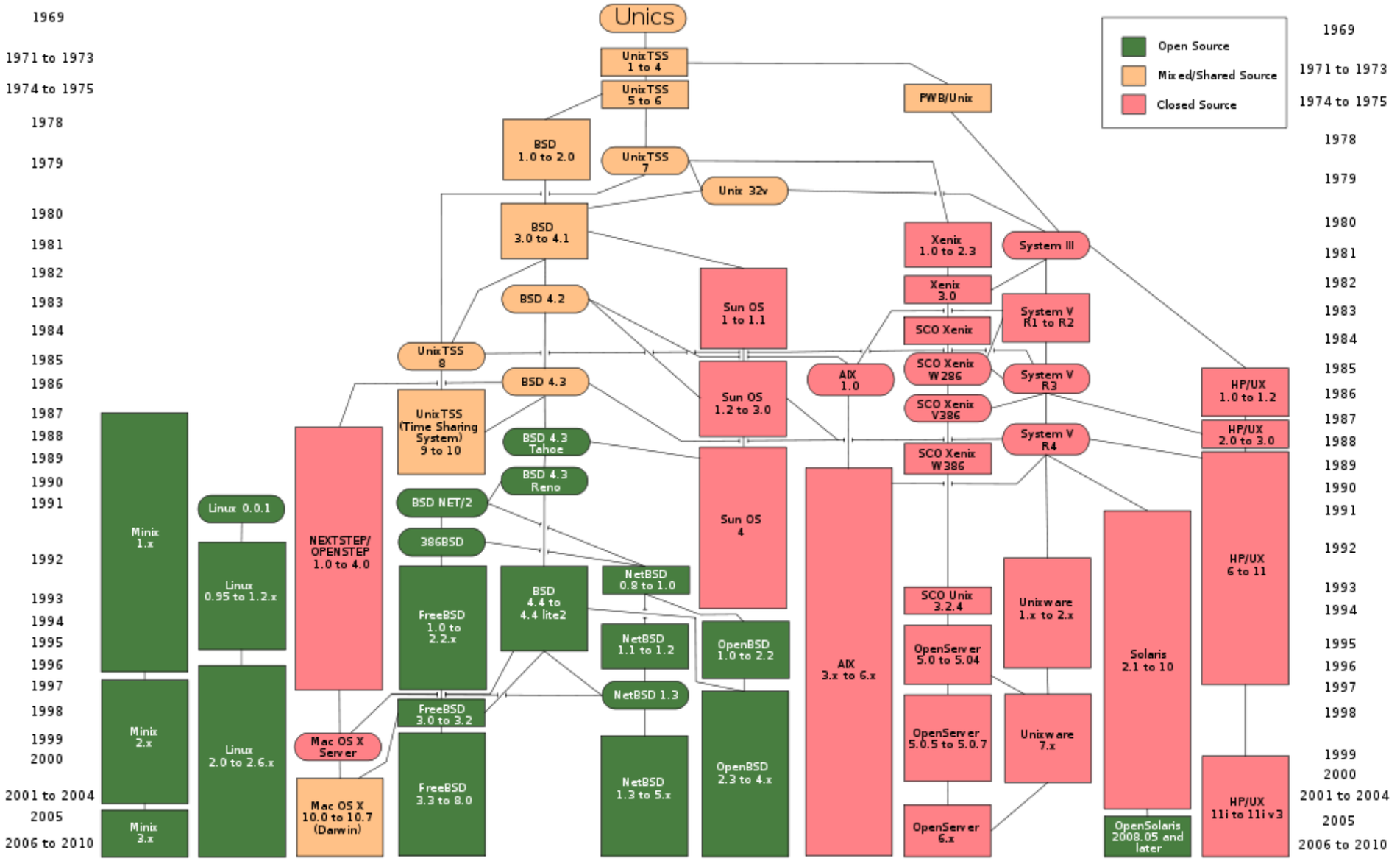
Aufgaben

- Ansteuern der Hardware
- Verwalten von Programme und Anwendungen
- Verteilen von Speicher, Ressourcen
- Verwalten von Benutzer, Rechte
- Verwalten von Daten, Dateisysteme
- Bereitstellung des API
- Anwendungen <-> BS <-> Hardware müssen zusammenpassen



UNIX

- Multitasking
  - mehrere Programme laufen gleichzeitig (echtes Multitasking)
  - Quasi-Multitasking auf Einprozessor-Systemen
  - Multitasking auf Mehrprozessor-Systemen
  - Scheduler verwaltet Ressourcen und teilt sie den verschiedenen Programmen zu
- Multiuser
  - mehrere Benutzer können ein System (gleichzeitig) benutzen
  - Rechteverwaltung, Isolation der Benutzer
- POSIX-API
- Serverplattform -> beste Netzeigenschaften
- BSD-Systeme, HP-UX (Hewlett-Packard), DG/UX (Data General), AIX (IBM), IRIX (Silicon Graphics), Solaris (Oracle), Mac OS X (Apple)



(Quelle: http://de.wikipedia.org/w/index.php?title=Datei:Unix_history-simple.svg&filetimestamp=20110807025810)

GNU – *GNU's Not Unix*

- Ziel: Freies Unix-Kompatibles Betriebssystem
- Ab 1983 by Richard Stallman

- POSIX-Userland
- Kernel: GNU/Hurd
  - *„GNU/Hurd ist die Zukunft, und wird es immer sein“*

Linux

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

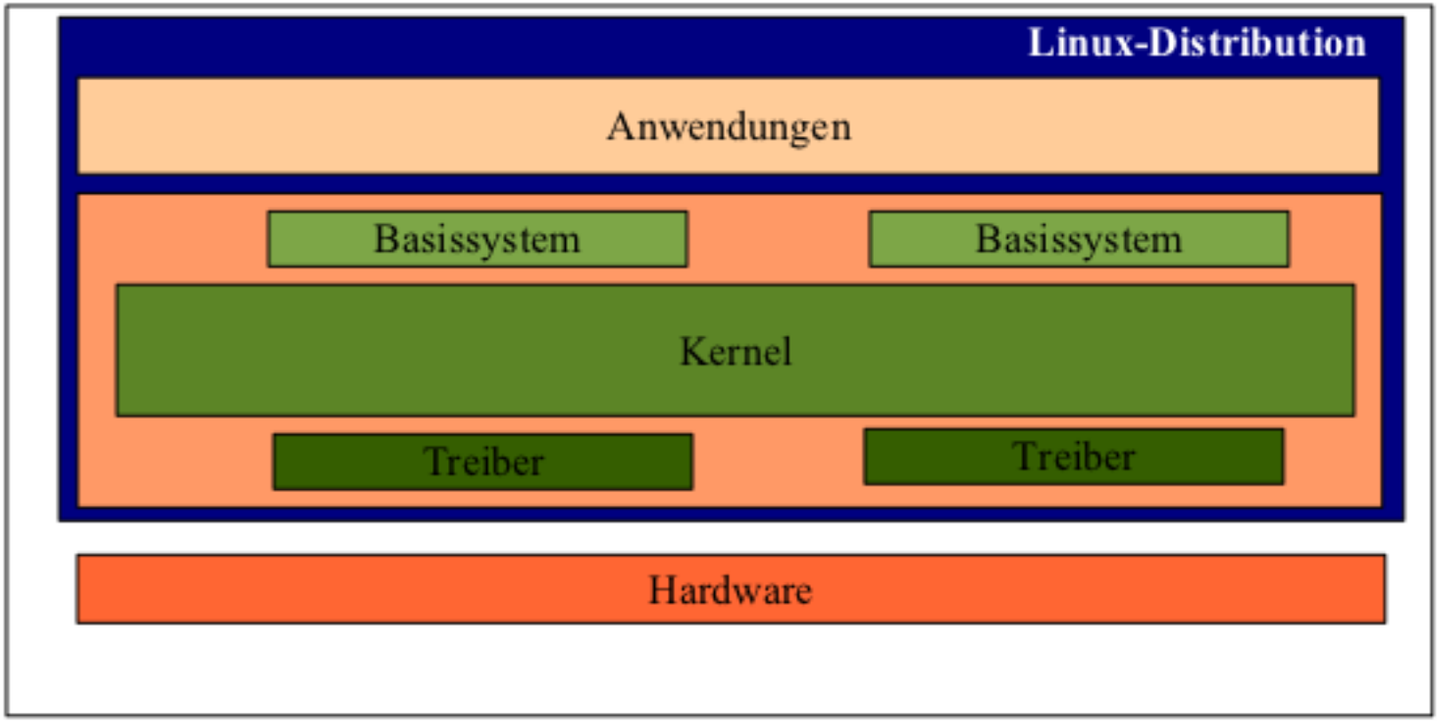
(Quelle: http://www.cs.cmu.edu/~awb/linux.history.html)

GNU/Linux

LIEGT IM QUELLTEXT VOR!!!

Distributionen

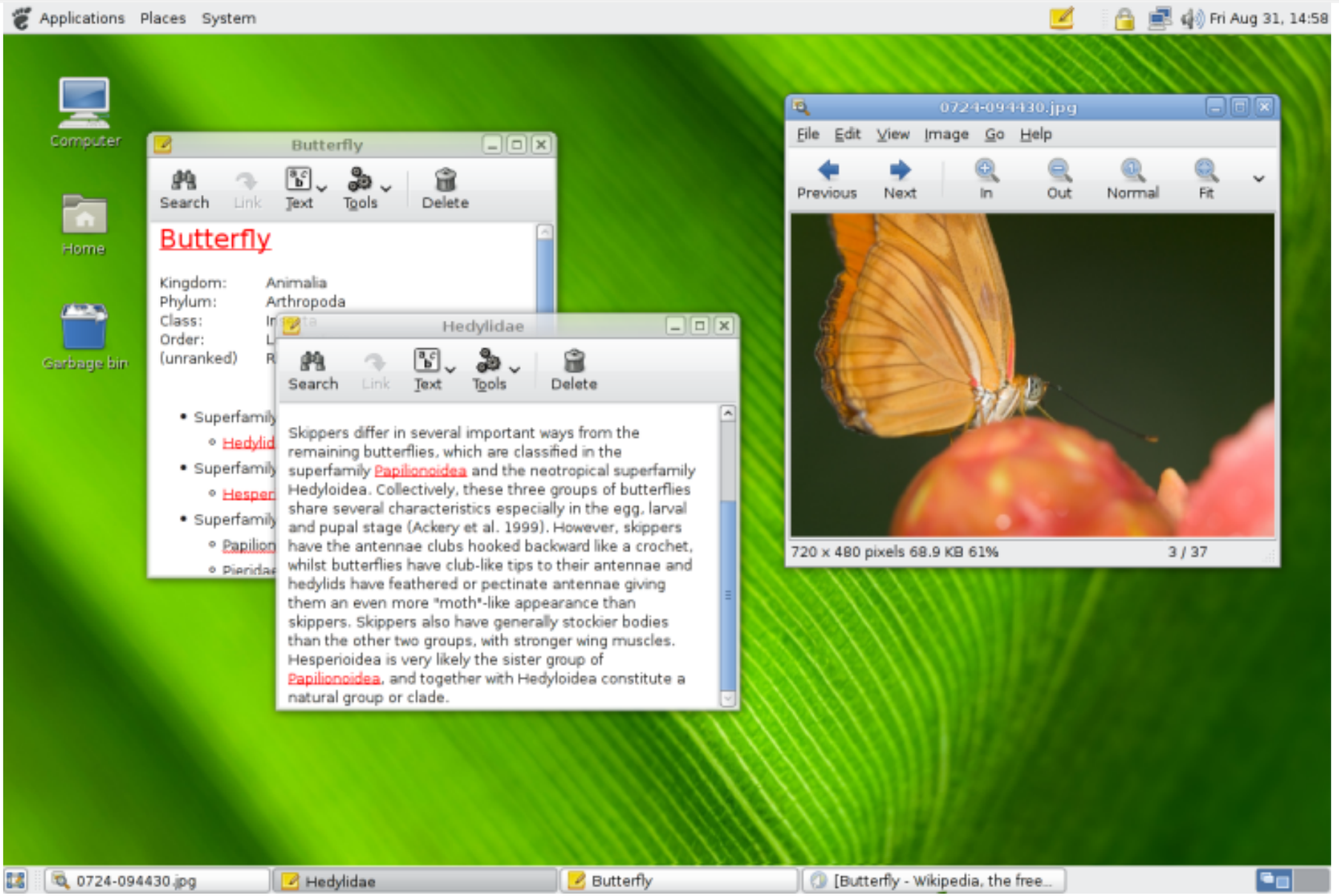
- Beinhaltet Tools für einheitliche und einfache Installation und Verwaltung
- Zusammenstellung von Kernel, Basissystem und Anwendungen
- Bsp.: Debian, Arch, Gentoo (Community), Ubuntu, SuSE, RedHat (Firmen), ...



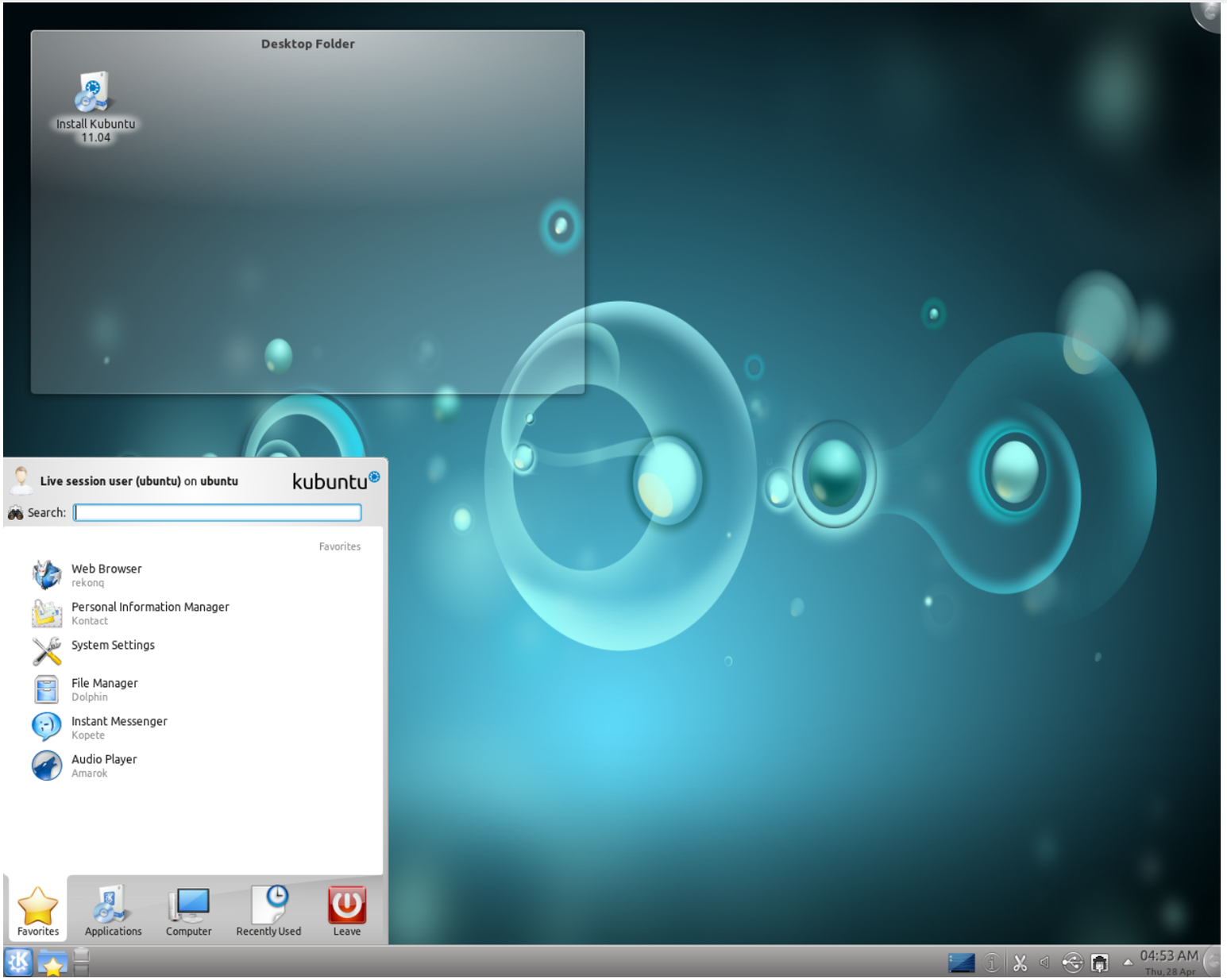
(Siehe: http://upload.wikimedia.org/wikipedia/commons/3/30/Gldt1010.svg)

GUI – Grafical User Interface

- Fenster, Buttons, Maus, …
- unter Linux „normales“ Programm -> austauschbar, nicht Teil des Betriebssystems -> nicht zwingend notwendig
- Unzählige zur Auswahl, probieren!
- unter Windows Teil des Betriebssystem

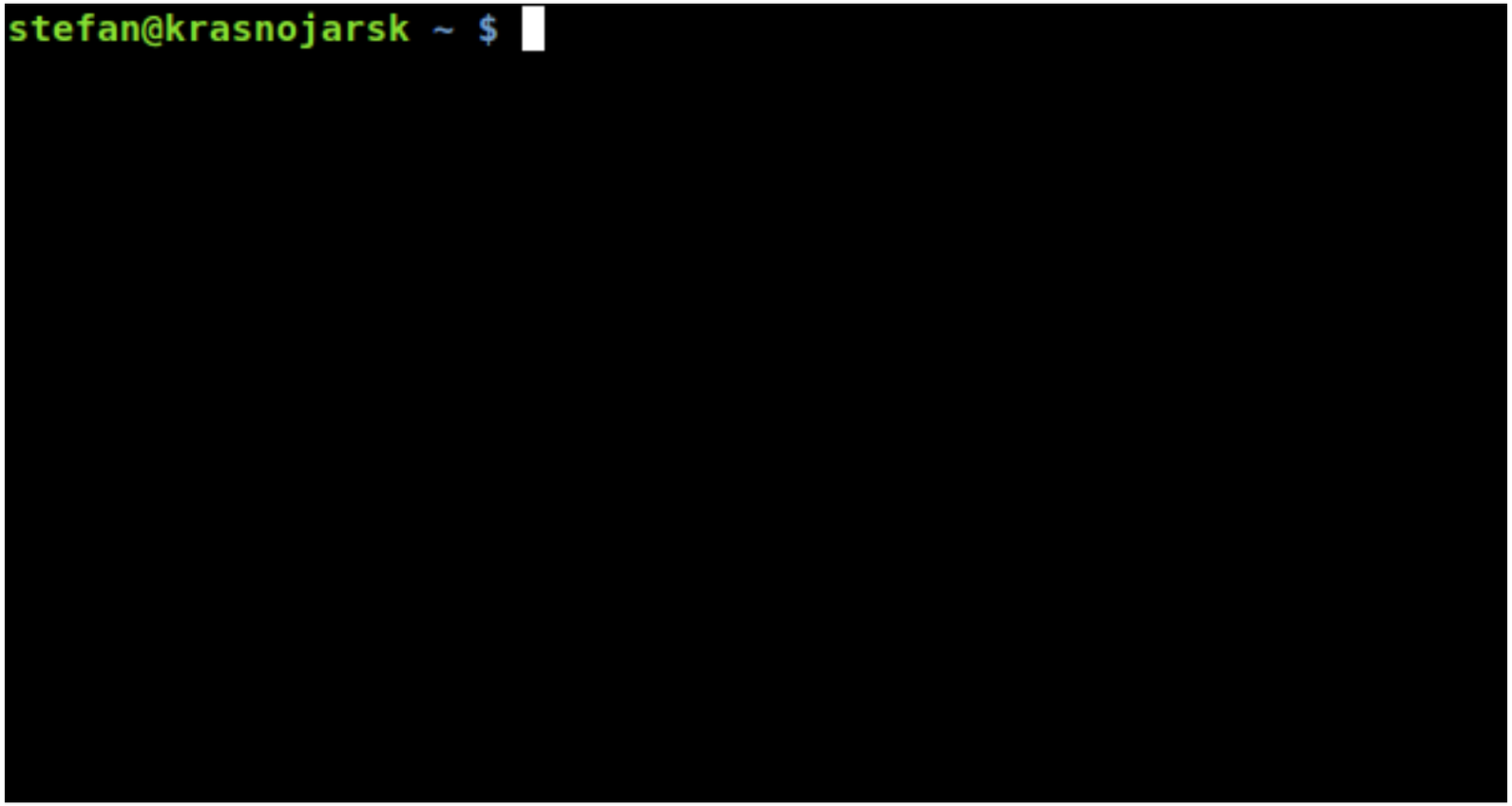


(Quelle: http://ca.wikipedia.org/wiki/Fitxer:Gnome-2.20-screenshot.png)



(Quelle: http://de.wikipedia.org/w/index.php?title=Datei:UbuntuNattyKDE.png&filetimestamp=20110429054345)

Shell



Exkurs: Benutzer, Gruppen

Benutzer (Multiuser)

- jeder Benutzer besitzt eigenes Passwort und muss sich am System anmelden
  - auch Auto-Login ist ein Login
- Benutzer gehört mindestens einer Gruppe an
- Benutzer / Gruppen verfügen über unterschiedliche Rechte am System
- jeder Benutzer verfügt über einen eigenen Ordner für persönliche Daten und Einstellungen
- Programme werden unter der Benutzerkennung und den Rechten des ausführenden Benutzers gestartet (*)
- Handlungen werden, ja nach Einstellung, protokolliert
- Systembenutzer für Systemprozesse / Dienste

Root

- Superuser, Gott
- darf alles, vollständiger Zugriff
- nur Root darf standardmäßig:
  - System herunterfahren, neu starten
  - Software installieren
  - Benutzer, Gruppen anlegen
  - Mounten *(Geduld, später mehr dazu)*
  - Gerätetreiber, Hardware starten
  - Systemkomponenten konfigurieren
  - in Ordnern anderer Benutzer schnüffeln
  - Protokolle lesen
- von Root ausgeführte Programme dürfen alles (siehe *)
  - → Sicherheitsrisiko → nie als Root arbeiten, wenn nicht unbedingt notwendig

Root werden

- als root anmelden (wenn Passwort gesetzt und bekannt)
  - schlechte Idee (siehe *)
- su (Shell)
- sudo (Shell)
- GUI-Komponenten (z.B. gksudo)

Exkurs: Verzeichnisbaum, Dateien

/		
/bin	binaries	Programme des Basissystems , GNU-System (bsp. ls, cat)
/boot	boot	Kernel, Ramdisks, Bootloader
/dev	devices	Gerätedateien
/etc	editable text configuraition	Systemweite Konfigurationsdateien
/home	home	Persönliche Daten und Einstellungen der Benutzer
/lib	libraries	Dynamische Bibliotheken des Basissystems
/media	media	Mountpoint für Wechselmedien (automatisch)
/mnt	mount	Mountpoint für Wechselmedien (manuell)
/opt	optional	Optionale (ClosedSource) Programme (bsp. flash)
/proc	processes	Systeminformationen, Kernel-Schnittstelle (virtuelles Dateisystem)
/root	root	Homeverzeichnis von Root
/sbin	system binaries	Programme Basissystem, nur von Root nutzbar (bsp. lsmod)
/srv	services	Nutzdaten von Diensten
/sys	system	siehe proc
/tmp	temporary	Temporäre Dateien
/usr	unix system resources	Programme, unter Paketverwaltung (bsp. firefox, openoffice)
/usr/local		Programme, nicht unter Paketverwaltung (kompiliert)
/var	variable	Veränderliche Daten
/var/log		Logdateien, Protokolle

Kein D:, E:, F:?

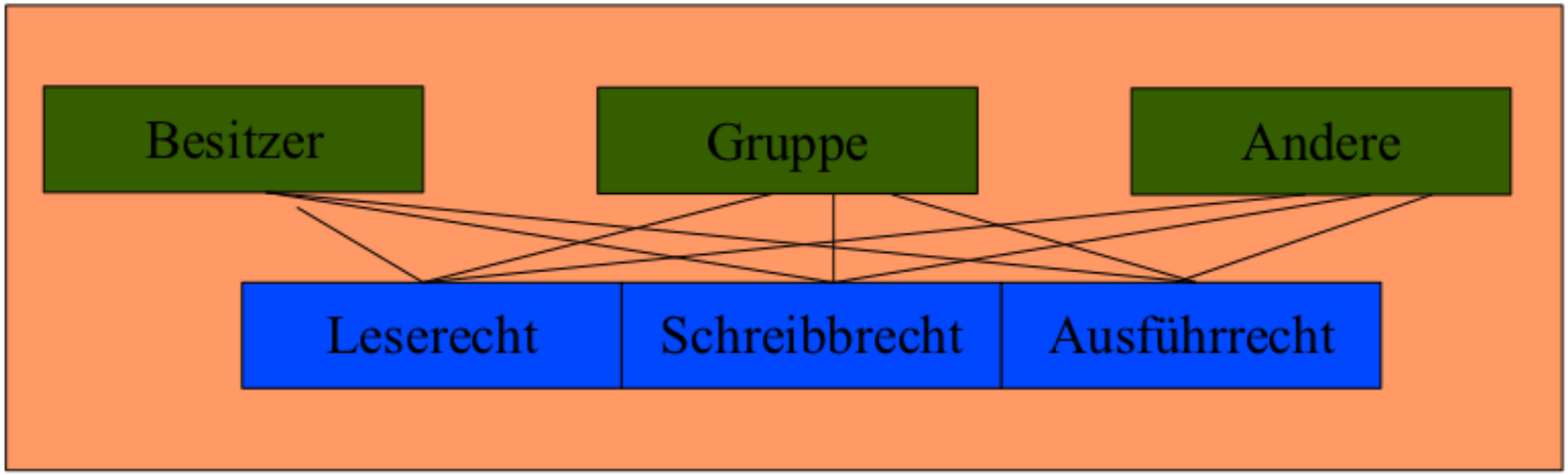
Wo ist meine CD / mein Stick???

Wechselmedien und Datenräger

- Verzeichnisbaum auf Datenträgern
  (Disketten, CD-ROMs, DVDs, USB, Festplatten-Partitionen, Netzwerk, gepackte Dateisysteme)
- wird an beliebiger Stelle im Verzeichnisbaum eingehangen (gemountet)
- Mountpoint kann beliebiger (leerer) Ordner sein
- verschiedene Dateisysteme und Datenträger werden transparent zusammengefügt
- Wechselmedien müssen vor Entnahme ausgehangen werden (unmount), sonst droht Datenverlust (Datencache)
- Moderne Distributionen mounten automatisch nach /media

Dateien und Rechteverwaltung

- Dateien haben keine Endung (z.B. kein .exe), der Typ wird automatisch erkannt
- Groß- und Kleinschreibung wird unterschieden (!!!)
- Dateien, die mit .irgendwas beginnen werden versteckt
- Dateien können ausführbar sein (→ Programme)
- Dateien, Ordner haben Besitzer und verfügen über Berechtigungen



Software verwalten

Alles easy:

Installieren, updaten, deinstallieren

→ Paketverwaltung der Distribution

Paketverwaltung

→ Software-Datenbank (Repository) + Tools für die Verwaltung dieser Software

Debian-Derivate

- dpkg / apt
- aptitude update, aptitude search FOO, aptitude install FOO, aptitude remove FOO, aptitude dist-upgrade
- GUI → Synaptic, …

RedHat-Derivate + SuSE

- rpm / yum / yast
- yum search FOO, yum install FOO, yum remove FOO, yum upgrade
- GUI → Synaptic, Yast, …

Arch-Linux + Slackware

- pacman
- pacman -Sy, pacman -Ss FOO, pacman -S FOO, pacman -Rs FOO, pacman -Su

Gentoo + Sabayone

- Portage
- emerge –-sync, emerge –-search FOO, emerge FOO, emerge –-unmerge FOO, emerge -avNDu world

Software ist nicht im Repository…

Versuch 1

- Repository finden und einbinden
  - ACHTUNG: Repo muss zur Distribution UND zur Version passen (!!!)
  - Ist das Repo vertrauenswürdig?
  - Tipp: So wenig Fremd-Repos wie irgend möglich nutzen
    - Dependency-Hell, Upgrades

Versuch 2

- aus den Quellen bauen
  - Quelltext besorgen und compilieren

Aus den Quellen bauen

Make

- Programm zum automatischen Compilieren
- Entwickler der Software beschreibt in einem Makefile wie die Compiler aufgerufen werden müssen
- einfacher Aufruf von make startet das korrekte compilieren der Software
- Make kann im Aufruf Bedingungen übergeben bekommen
  (make, make install, make dep, make clean, ...)

Vorbereitungen

- Compiler + Tools (gcc, g++, make, automake, ...) installieren
- Header von abhängiger Software installieren (Dev-Pakete)

Software installieren, allgemeiner Weg

1. Quellcode besorgen (Internet)
2. Auspacken mit tar
3. System-Abhängigkeiten vorkonfigurieren mit
   ./configure (beigelegtes Programm, ./configure –-help bietet Hilfe)
4. (Abhängigkeiten von Programmen überprüfen, ggf. dev-Pakete installieren)
5. Compilieren mit make
6. Binärdateien Installieren mit make install

ACHTUNG: Allgemein anerkannter Weg, wird aber vom Programmierer der Software bestimmt. Die Installation kann auch ganz anders verlaufen, immer zuerst in der entsprechenden Dokumentation nachlesen.

Es gilt RTFM!!!

Zusammenfassung:

tar -xvjf bzw. tar -xvzf

./configure

make

make install

ACHTUNG: Selbst gebaute Software ist nicht unter Paketverwaltung → keine automatischen Updates

Shell

man-Pages

manual -> Hilfe, Benutzerhinweise, Programmbeschreibung

DEMO: man ls, man man, Parameter, Optionen

Cheat Sheet

http://files.fosswire.com/2007/08/fwunixref.pdf

Pfade

- pwd
- absolut
  - /home/tux/Desktop/
  - /etc/fstab
  - /usr/bin/firefox
- relativ
  - Desktop
  - ./Desktop
  - ../../etc/fstab
  - cd .. (nicht cd.. !!!)

Piping

- Ausgabe eines Programms -> Eingabe eines zweiten Programms
  (Grund für z.T. seltsame Standardausgabe von Linux-Programmen)
- Symbol für Pipe: |

DEMO: ls /etc | wc

Datenströme

- Geräte (Tastatur, Maus, Bildschirm, ...) werde als Datei aufgefasst
- Tastatur = stdin (Standardeingabe), Bildschirm = stdout (Standardausgabe)
- kann verändert werden, z.B. Drucker = stdout
- BEFEHL < DATEI Datei wird zu stdin
- BEFEHL > DATEI Datei wird zu stdoud

DEMO:

echo ''hallo ein schöner tag'' > test
sort < test > testsortiert
cat > test

- > löscht Inhalt der Datei, >> hängt an Datei an

Shellscripts

- Shell kann als Befehlsinterpreter arbeiten
- Linux-Kommandos können zu leistungsfähigen Programmen (Shellscripts) verknüpft werden
- Systemverwaltung kann mit Shellscripts automatisiert werden
- Beim Systemstart werden zum Initialisieren Shellscripts aufgerufen
- Kommandos werden nacheinander ausgeführt (vgl. MS-DOS Batch)

DEMO:

```
#!/bin/sh
#Ein kleines Demoscript
/bin/echo '' ein kleinses Demoscript''
/bin/ls -l
/bin/echo '' Das ist der Inhalt vom Verzeichnis''
/bin/pwd
```

Shellscripts

- Shellscripts können bzw. verfügen über:
  - Variablen
  - Bedingte Anweisungen (wenn, dann)
  - Schleifen (Wiederholungen)
  - Funktionen

Scripts ausführen

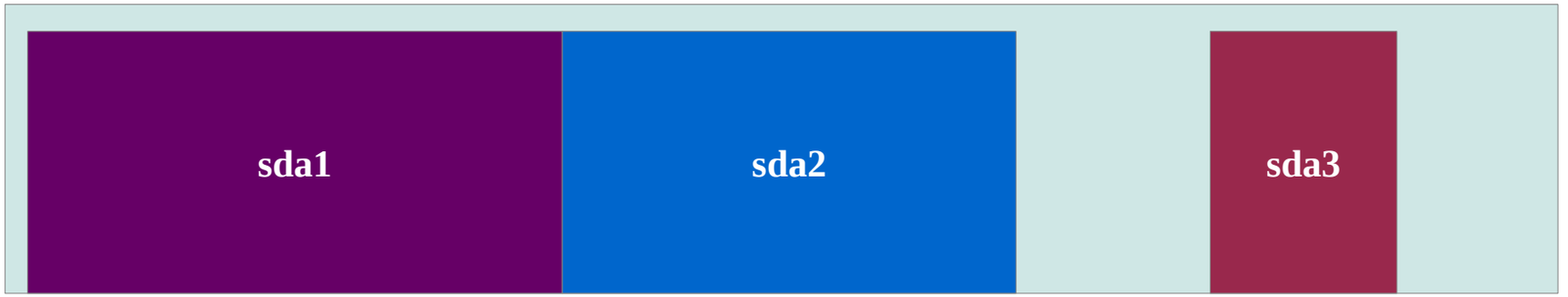
- /bin/sh SHELLSCRIPT

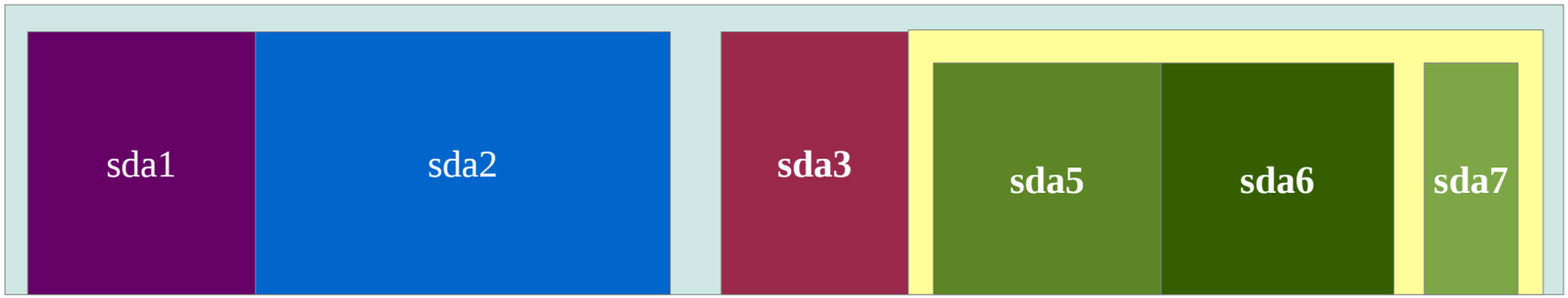
oder

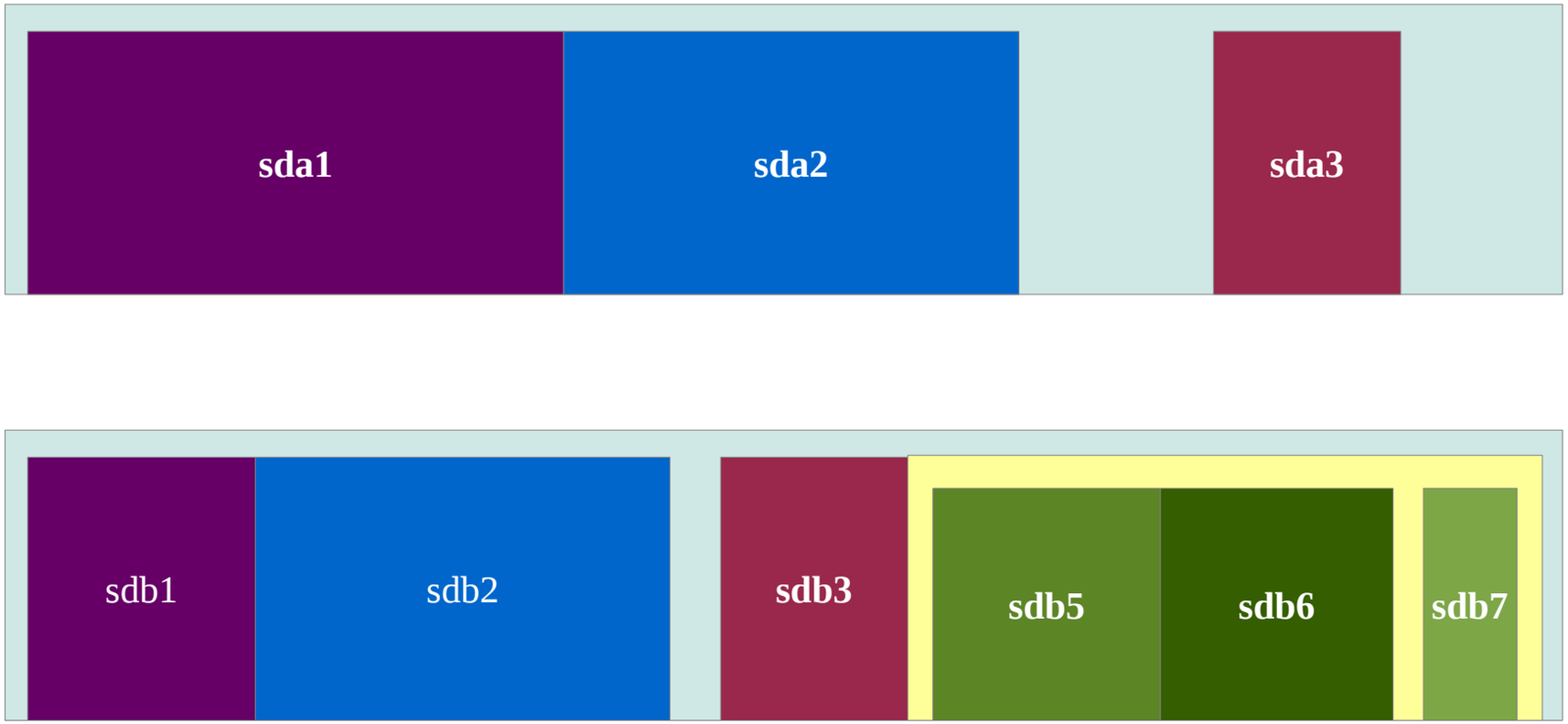
- chmod +x SHELLSCRIPT
- ./SHELLSCRIPT

Partitionen

Festplatte


sda1
sda2
sda3


sda1
sda2
sda3
sda5
sda6
sda7


sda1
sda2
sda3
sdb1
sdb2
sdb3
sdb5
sdb6
sdb7

Dateisysteme

- Beschreiben die Struktur der Daten auf einem Datenträger
- Anlegen durch Formatierung einer Partition
  - mkfs.DATEISYSTEM /dev/sdXY
  - Bsp: mkfs.ext4 /dev/sda5
- Features könne sein: Rechteverwaltung, Journaling, Snapshots, Redundanz, ACL, Quota, ...

ext3	Linux Standard-Dateisystem	2TB / 32TB
ext4	Linux Standard-Dateisystem neu	16TB / 1EB
xfs	Leistungsfähiges Dateisystem für Server	8EB / 8EB
fat32	Windows ab 95b viele Sticks / Speicherkarten	4GB / 8TB
ntfs	Windows ab NT	16EB / 16EB
zfs	Hochleistungsdateisystem von Sun	16EB / 16EB
btrfs	Hochleistungsdateisystem, zukünftiges Linux Standard-Dateisystem	16EB / 16EB

Swap

- Auslagerungsspeicher
- kein RAM (!!!)
  - nein, auch kein virtueller RAM
- ungenutzte Daten aus dem RAM werden auf eine Partition ausgelagert
  - → mehr RAM steht zur Verfügung
  - ABER! Auslagerungsprozesse brauchen viel Zeit

- mkswap /dev/sda2

Partitionslayout

(Stichworte: Verzeichnisbaum, mounten)

- Swap
  - 2*RAM, höchstens 1GB
- /
  - zu erwartende Größe von Linux
  - 10GB – 15GB
- /home
  - so viel wie möglich / noch Platz
- /boot
  - auf Computern, die nicht von großen / erweiterten Partitionen booten
  - 256MB-512MB
  - ext2

- /tmp
- /var
- /var/log
- …
- siehe: /etc/fstab

Stichworte

Logical Volume Manager / lvm2

Software RAID / mdadm

Verschlüsselung / Luks

Netzwerkdateisysteme / nfs, OpenAFS

Linux-Systemstart

Bootstrap

- BIOS initialisiert System
  - springt zu
- Bootmanager im reservierten Bootsektor der Festplatte
  - entpackt und startet
- Kernel mit Prozessnummer 0
  - startet Prozess
- init mit Prozessnummer 1
  - liest und durchläuft
- Runlevels (rc-Scripte)

Bootmanager

- Programm im Bootsektor der Festplatte
- liegt an vordefinierter Adresse und wird vom Bios angesprungen
- Startet den Linux-Kernel bzw. andere installierte Betriebssysteme

Grub / Grand Unified Bootloader

- Grub legacy, wird über /boot/grub/menu.list konfiguriert
- Grub 2, wird über /etc/default/grub installiert (debian-derivate)

Linux und Windows

Variante 1

- extra Partitionen (Windows will unbedingt auf sda1)
- grub Bootloader
- Auswahl beim Start des Computers, nicht gleichzeitig
- bei bereits installiertem Windows von vielen Linux-Distributionen automatisch unterstützt
  - trotzdem vorher Backup machen!

Variante 2

- Virtualisierung
  - VirtualBox, vmWare
- gleichzeitiger Betrieb möglich
- hohe Hardware-Anforderungen

Wie weiter?

Links:

http://openbook.galileocomputing.de/unix_guru/
http://openbook.galileocomputing.de/linux/
http://openbook.galileocomputing.de/ubuntu/

Slides:

http://download.terminal21.de/workshop/pinguin_kindergarten/pinguin_slides.pdf

Kontakt:

http://www.terminal21.de
stefan@terminal21.de