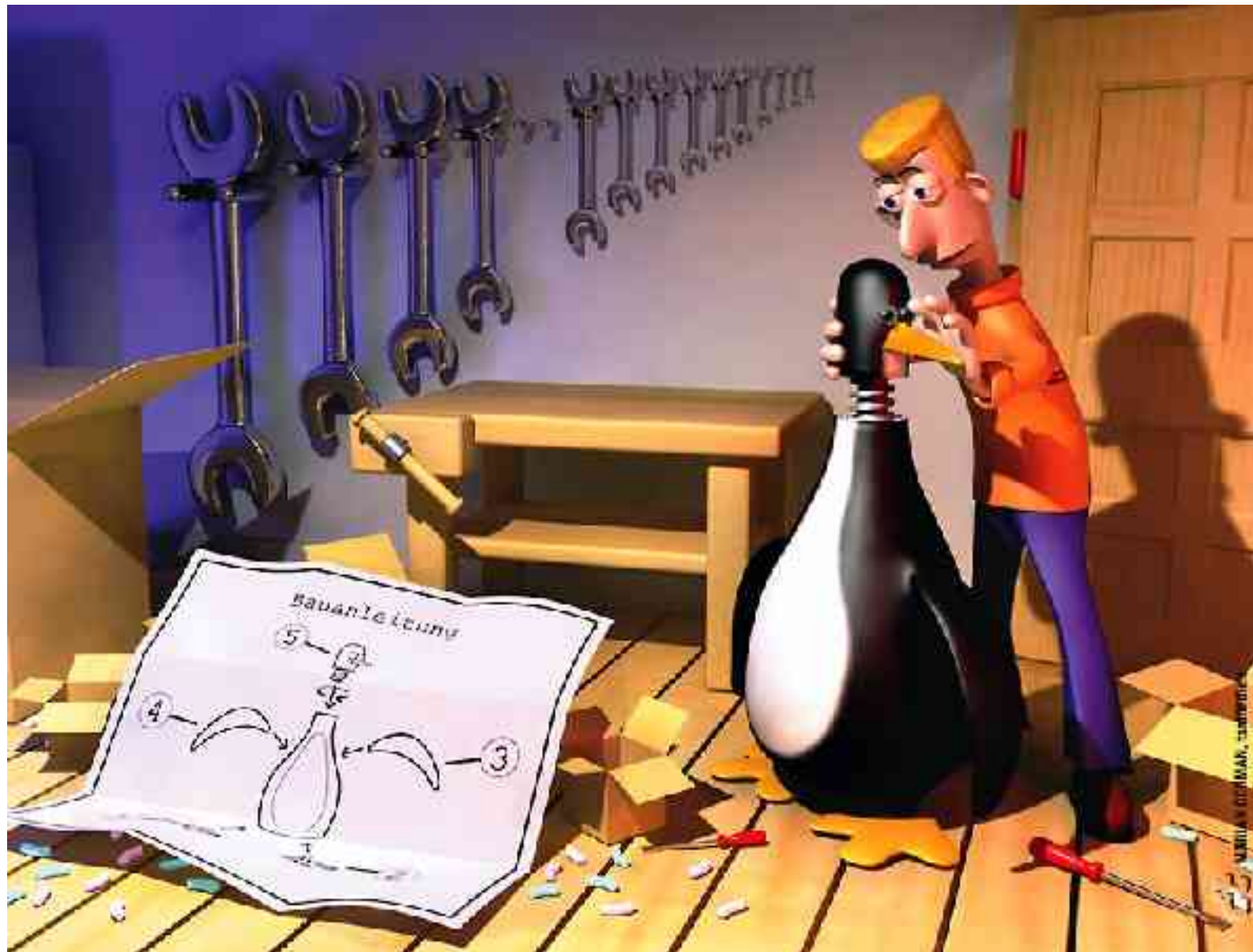


Linux selber bauen



- **Warum Linux selber bauen?**
- Distributionen
- Voraussetzungen
- Vorgehensweisen
- Aufbau des Linux Systems

Linux selber bauen - Warum Linux selber bauen?

- System auf die Hardware optimieren
- neuste Versionen
- selbst bestimmen, was installiert wird
- Programme auf optionale Funktionalität optimieren
- Linux von innen im Detail kennenlernen

Linux selber bauen - Übersicht

- Warum Linux selber bauen?
- **Distributionen**
- Voraussetzungen
- Vorgehensweisen
- Aufbau des Linux Systems

Linux selber bauen - Distributionen

- LFS – Linux From Scratch
<http://www.linuxfromscratch.org/>
- FreeBSD
<http://www.freebsd.org/>
- Gentoo
<http://www.gentoo.org/>



Linux selber bauen - Übersicht

- Warum Linux selber bauen?
- Distributionen
- **Voraussetzungen**
- Vorgehensweisen
- Aufbau des Linux Systems

Linux selber bauen - Voraussetzungen

- Festplattenplatz (ca. 10 GB)
- Ein laufendes Wirt-System (Linux, BSD...)
- Compiler (gcc)
- C-Bibliotheken (glibc)
- Systemtools (binutils, autoconf, automake, gettext, tar, bzip2...)
- Quellen der Pakete (Internetzugang)
- Zeit (für Desktopsystem ca. 2 Tage)

Linux selber bauen - Übersicht

- Warum Linux selber bauen?
- Distributionen
- Voraussetzungen
- **Vorgehensweisen**
- Aufbau des Linux Systems

- LFS:
 - Wirt-System starten
 - Partitionierung vornehmen
 - Dateisysteme anlegen
 - Dateisysteme mounten
 - Verzeichnisstruktur anlegen
 - Benötigte Pakete als static kompilieren
 - chroot ins neue System
 - Benötigte Pakete als shared kompilieren
 - Kernel und Bootloader installieren
 - Neustart
 - gewünschte Programme installieren

Linux selber bauen - Vorgehensweisen

- FreeBSD:
 - FreeBSD von CD installieren
 - Quellen installieren
 - installiertes System starten
 - cvsup ausführen
 - `/usr/src/sys/i386/conf/GENERIC` anpassen
 - `make buildkernel / make installkernel`
 - `/etc/make.conf` anpassen
 - `make buildworld / make installworld`
 - mergemaster ausführen
 - portupgrade installieren
 - gewünschte Programme mit portinstall installieren



- Gentoo:
 - Wirt-System starten
 - Gentoo Installations-CD
 - beliebiges UNIX
 - Netzwerk aktivieren
 - Partitionierung vornehmen
 - Dateisysteme anlegen
 - Dateisysteme mounten
 - gewünschtes Grundsystem entpacken
 - Stage1
 - Stage2
 - Stage3
 - chroot



- Gentoo:
 - emerge sync ausführen
 - /etc/make.conf anpassen
 - ggf. bootstrap starten
 - emerge system
 - gewünschter logger und cron installieren
 - Kernel installieren
 - Neustart
 - emerge world
 - gewünschte Pakete mit emerge installieren
 - Runlevel und Dienste konfigurieren



Linux selber bauen - Übersicht

- Warum Linux selber bauen?
- Distributionen
- Voraussetzungen
- Vorgehensweisen
- **Aufbau des Linux Systems**

- **Bootvorgang**
- Kernel
- Runlevels
- SysVinit
- Dienste
- Login

Linux selber bauen - Bootvorgang

- Linux Loader (LILO / GRUB)
- Kernel
- SysVinit
- Runlevel 2
- Runlevel 3
- Login (getty / Displaymanager)

- Bootvorgang
- **Kernel**
- Runlevels
- SysVinit
- Dienste
- Login

- Der Linux Kernel:
 - Geräte-Treiber
 - Dateisysteme
 - Schnittstellen zur Hardware (/dev)
 - Schnittstellen zum Netzwerk
 - Systeminformationen (/proc)
 - startet den init-Prozess
 - verwaltet Speicher
 - steuert Prozesse

- monolithischer Kernel
 - alle Funktionen die zum Start des Rechners gebraucht werden
- Module
 - könne nach Belieben nachgeladen werden
 - externe (binäre) Module von Hardware-Hersteller

- Bootvorgang
- Kernel
- **Runlevels**
- SysVinit
- Dienste
- Login

SysVinit kontrolliert Runlevels
Wechsel mit `init x` (x =Runlevel)

- Runlevel 0
Halt
- Runlevel 1 / Runlevel S
Single-User-Mode ohne Netzwerk
- Runlevel 2
System ohne Netzwerk
- Runlevel 3
System mit Netzwerk
- Runlevel 6
Reboot

- Bootvorgang
- Kernel
- Runlevels
- **SysVinit**
- Dienste
- Login

- init ist der "Vater" aller Prozesse
- init kann nicht beendet werden
- init wechselt die Runlevel
- init startet und beendet die für das Runlevel konfigurierten Dienste

Linux selber bauen - SysVinit

```
/etc/inittab: # Default runlevel.
id:3:initdefault:

# System initialization, mount local filesystems, etc.
si::sysinit:/sbin/rc sysinit

# Further system initialization, brings up the boot runlevel.
rc::bootwait:/sbin/rc boot

10:0:wait:/sbin/rc shutdown
11:S1:wait:/sbin/rc single
12:2:wait:/sbin/rc nonetwork
13:3:wait:/sbin/rc default
14:4:wait:/sbin/rc default
15:5:wait:/sbin/rc default
16:6:wait:/sbin/rc reboot
#z6:6:respawn:/sbin/sulogin

# TERMINALS
c1:12345:respawn:/sbin/agetty 38400 tty1 linux
c2:12345:respawn:/sbin/agetty 38400 tty2 linux
c3:12345:respawn:/sbin/agetty 38400 tty3 linux
c4:12345:respawn:/sbin/agetty 38400 tty4 linux
c5:12345:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux

# What to do at the "Three Finger Salute".
ca:12345:ctrlaltdel:/sbin/shutdown -r now

x:a:once:/etc/X11/startDM.sh
# End of /etc/inittab
```


jedes Runlevel enthält seine eigenen Dienste

- `/etc/init.d/`
enthält Startskripts aller Dienste
- `/etc/conf.d/`
enthält die Konfiguration des
entsprechenden Startskripts
- `/etc/runlevels/boot`
enthält Startskripts für Runlevel 1
- `/etc/runlevels/nonetwork`
enthält Startskripts für Runlevel 2
- `/etc/runlevels/default`
enthält Startskripts für Runlevel 3

Linux selber bauen - Aufbau des Linux Systems

- Bootvorgang
- Kernel
- Runlevels
- SysVinit
- **Dienste**
- Login

Linux selber bauen - Dienste

boot:

insgesamt 7

lrwxrwxrwx	1	root	root	20	2. Mai 2003	bootmisc -> /etc/init.d/bootmisc
lrwxrwxrwx	1	root	root	19	2. Mai 2003	checkfs -> /etc/init.d/checkfs
lrwxrwxrwx	1	root	root	21	2. Mai 2003	checkroot -> /etc/init.d/checkroot
lrwxrwxrwx	1	root	root	17	2. Mai 2003	clock -> /etc/init.d/clock
lrwxrwxrwx	1	root	root	23	2. Mai 2003	consolefont -> /etc/init.d/consolefont
lrwxrwxrwx	1	root	root	18	4. Mai 2003	hdparm -> /etc/init.d/hdparm
lrwxrwxrwx	1	root	root	20	2. Mai 2003	hostname -> /etc/init.d/hostname
lrwxrwxrwx	1	root	root	19	2. Mai 2003	keymaps -> /etc/init.d/keymaps
lrwxrwxrwx	1	root	root	22	2. Mai 2003	localmount -> /etc/init.d/localmount
lrwxrwxrwx	1	root	root	19	2. Mai 2003	modules -> /etc/init.d/modules
lrwxrwxrwx	1	root	root	18	2. Mai 2003	net.lo -> /etc/init.d/net.lo
lrwxrwxrwx	1	root	root	21	2. Mai 2003	rmnologin -> /etc/init.d/rmnologin
lrwxrwxrwx	1	root	root	18	2. Mai 2003	serial -> /etc/init.d/serial
lrwxrwxrwx	1	root	root	19	2. Mai 2003	urandom -> /etc/init.d/urandom

default:

insgesamt 7

lrwxrwxrwx	1	root	root	17	5. Mai 2003	acpid -> /etc/init.d/acpid
lrwxrwxrwx	1	root	root	17	1. Jul 2003	cupsd -> /etc/init.d/cupsd
lrwxrwxrwx	1	root	root	15	4. Mai 2003	gpm -> /etc/init.d/gpm
lrwxrwxrwx	1	root	root	18	4. Mai 2003	hdparm -> /etc/init.d/hdparm
lrwxrwxrwx	1	root	root	19	14. Nov 20:21	hotplug -> /etc/init.d/hotplug
lrwxrwxrwx	1	root	root	16	29. Jul 2003	irda -> /etc/init.d/irda
lrwxrwxrwx	1	root	root	17	2. Mai 2003	local -> /etc/init.d/local
lrwxrwxrwx	1	root	root	20	2. Jul 2003	net.eth0 -> /etc/init.d/net.eth0
lrwxrwxrwx	1	root	root	20	2. Mai 2003	netmount -> /etc/init.d/netmount
lrwxrwxrwx	1	root	root	16	4. Mai 2003	sshd -> /etc/init.d/sshd
lrwxrwxrwx	1	root	root	20	3. Mai 2003	sysklogd -> /etc/init.d/sysklogd
lrwxrwxrwx	1	root	root	17	3. Mai 2003	vcron -> /etc/init.d/vcron
lrwxrwxrwx	1	root	root	15	3. Mai 2003	xdm -> /etc/init.d/xdm

nonetwork:

insgesamt 1

lrwxrwxrwx	1	root	root	18	4. Mai 2003	hdparm -> /etc/init.d/hdparm
lrwxrwxrwx	1	root	root	17	2. Mai 2003	local -> /etc/init.d/local

- boot
 - clock
Uhreinstellungen setzen (Zeitzone / lokale Zeit)
 - checkfs / checkroot
Dateisysteme prüfen
 - consolefont
Schriftart der Konsole laden
 - hdparm
Festplatten-Parameter setzen (DMA-Modus etc.)
 - hostname
Rechnername setzen

- boot, Fortsetzung...
 - keymaps
Tastatur-Layout laden
 - localmount
lokale Dateisysteme einhängen
 - modules
Kernel-Module laden
 - net.lo
Loopback Netzwerk starten
 - serial
Serielle Schnittstelle initialisieren

- nonetwork
 - local
Programme ausführen die beim Start von Linux gestartet werden sollen
 - hdparm
Festplatten-Parameter setzen (DMA-Modus etc.)

- default
 - acpid
ACPI Daemon
 - cupsd
Druckserver
 - gpm
Maus-Unterstützung für die Konsole
 - hdparm
Festplatten-Parameter setzen (DMA-Modus etc.)
 - hotplug
Geräte die während dem Betrieb eingesteckt werden initialisieren
 - irda
Infrarot-Schnittstelle initialisieren

Linux selber bauen - Dienste

- default, Fortsetzung...
 - local
Programme ausführen die beim Start von Linux gestartet werden sollen
 - net.eth0
Netzwerkkarte aktivieren
 - netmount
Netzwerkdateisysteme einhängen
 - sshd
Secure-Shell Server starten
 - sysklogd
Dienst der Systemmeldungen in Logfiles schreibt
 - vcron
Dienst der zu bestimmten Zeiten Jobs startet
 - xdm
grafischer Loginmanager

Linux selber bauen - Aufbau des Linux Systems

- Bootvorgang
- Kernel
- Runlevels
- SysVinit
- Dienste
- **Login**

- **lokales Login**
- remote Login
- grafisches Login

- getty

Wird von init gestartet und wartet auf einer Konsole auf ein Login.

Ein getty kann sowohl auf einem Terminal als auch einer seriellen Konsole warten.

>> Modem-Zugriff

- lokales Login
- **remote Login**
- grafisches Login

- telnet

unverschlüsselter Zugriff auf eine Shell via Netzwerk.

NACH MÖGLICHKEIT NICHT VERWENDEN

- ssh

verschlüsselter Zugriff auf eine Shell via Netzwerk.

ssh-Client für Windows: putty

Linux selber bauen - Login

- lokales Login
- remote Login
- **grafisches Login**

- xdm

bei X enthaltener Displaymanager mit grafischem Login

- entsprechende Replacements:
 - kdm
KDE Loginmanager
 - gdm
GNOME Loginmanager

- Nächstes Mal:
 - Installation Gentoo Schritt für Schritt
 - LILO und Kernel Konfiguration
 - emerge und qpkg leichtgemacht
 - USE- und C-Flags im Detail

Vielen Dank für Eure Aufmerksamkeit

