

# Betriebssysteme

WS 2014/15

**Hans-Georg Eßer**  
Dipl.-Math., Dipl.-Inform.

**Foliensatz E:**

- Paketverwaltung

v1.1, 2014/11/21

# Übersicht: BS Praxis und BS Theorie

Einführung

Shell

Dateiverwaltung

Filter

C-Compiler

Prozesse / Jobs

Threads

Interrupts

System Calls

Scheduler / Prioritäten

Synchronisation

Speicherverwaltung

Software-Verwaltung

Partitionen

Auskunft

User, Gruppen, Rechte

Folien E

# Software-Verwaltung

- Verschiedene Linux-Distributionen verwenden unterschiedliche Software-Paketformate:
  - *\*.deb* – Debian-Paketformat (z. B. Debian, Ubuntu, Knoppix)
  - *\*.rpm* – Red-Hat-Paketformat (z. B. OpenSuse, Fedora / Red Hat, Mandriva)
  - weitere (weniger verbreitete Formate), z. B. *\*.txz* (Slackware, xz-komprimiertes *tar*-Archiv)
- Pakete enthalten die eigentliche Software und „Anleitungen“ für (De-) Installation

- Die beiden „großen“ Paketformate sind
  - das Debian-Format (*\*.deb*) und
  - das RPM-Format (*\*.rpm*, Red Hat Package Manager)
- Software-Verwaltung mit *deb/rpm*-Paketen findet immer auf zwei Ebenen statt:
  - auf Ebene einzelner Pakete; Tools:
    - dpkg (Debian Package Manager)
    - rpm (Red Hat Package Manager)
  - auf Ebene von Repositories; Tools:
    - APT (Advanced Packaging Tool / Debian)
    - YUM (Yellowdog Updater, Modified / Red Hat, Fedora)



- Funktionen auf Paketebene
  - Paket installieren / deinstallieren / aktualisieren
  - Paketinhalt anzeigen
  - erkennt → Abhängigkeiten und → Konflikte, verweigert evtl. (De-) Installation oder Update
  - zu installierter Datei das zugehörige Paket finden
  - Installation / Update benötigt immer eine konkrete Paketdatei

- Funktionen auf Repository-Ebene (1)
  - Ein Repository (Repo) ist eine Paketquelle, z. B.
    - Installations-CD/DVD
    - Verzeichnis auf einem Web- oder FTP-Server
    - lokales Verzeichnis
  - Repos haben ein Inhaltsverzeichnis mit den wichtigsten Metadaten zu allen vorhandenen Paketen
  - Repos für jede Distributions-Version separat

- Funktionen auf Repository-Ebene (2)
  - Anlegen / Aktualisieren / Entfernen von Repos (→ parallele Nutzung mehrerer Repos möglich)
  - Suche nach Paketen in allen eingetragenen Repos
  - Installation mit automatischem Auflösen von Abhängigkeiten
  - Komplett-Upgrade auf neue Version einer Linux-Distribution
  - Drittanbieter-Repos (z. B. für Multimedia-Pakete, Treiber vom Hardware-Hersteller oder proprietäre Software)

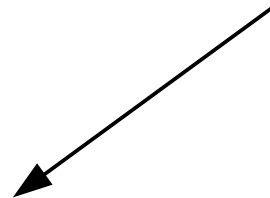


- Paket installieren:



```
[root@redhat:~]# ls
testpaket-1.2.3.i386.rpm
[root@redhat:~]# rpm -ivh testpaket-1.2.3.i386.rpm
Preparing... ##### [100%]
 1:testpaket ##### [100%]
[root@redhat:~]# _
```

Optionen `-vh` erzeugen die Ausgabe



```
[root@debian:~]# ls
joe_3.5-1.1_amd64.deb
[root@debian:~]# dpkg -i joe_3.5-1.1_amd64.deb
Wähle vormals abgewähltes Paket joe. (Lese Datenbank ...
113154 Dateien und Verzeichnisse sind derzeit installiert.)
Entpacke joe (aus ../archives/joe_3.5-1.1_amd64.deb) ...
Richte joe ein (3.5-1.1) ...
[root@debian:~]# _
```

- Paket aktualisieren (Update):



```
[root@redhat:~]# rpm -ivh dhcpd-3.2.3-66.69.1.i586.rpm
Preparing... ##### [100%]
file /sbin/dhcpd from install of dhcpd-3.2.3-66.69.1.i586 conflicts with file from package dhcpd-3.2.3-65.1.i586
```

```
[root@redhat:~]# rpm -Uvh dhcpd-3.2.3-66.69.1.i586.rpm
Preparing... ##### [100%]
1:dhcpd ##### [100%]
```



```
[root@debian:~]# dpkg -i joe_3.5-1.1_amd64.deb
```

(Update bei Debian auch über Installationsoption -i)

- Paket löschen: benötigt nur Paketnamen, nicht die Versionsnummer



```
[root@redhat:~]# rpm -e testpaket
```

e = erase



```
[root@debian:~]# dpkg -r joe
(Lese Datenbank ... 113175 Dateien und
Verzeichnisse sind derzeit installiert.)
Entferne joe ...
```

r = remove

oder

```
[root@debian:~]# dpkg -P joe
(Lese Datenbank ... 113175 Dateien und
Verzeichnisse sind derzeit installiert.)
Entferne joe ...
Lösche Konfigurationsdateien von joe ...
```

P = purge

- Paketinhalte:



```
[root@redhat:~]# rpm -ql rpm
/bin/rpm
/etc/init.d/rpmconfigcheck
/etc/rpm
/usr/bin/gendiff
/usr/bin/rpm2cpio
/usr/bin/rpmbuild
/usr/bin/rpmdb
...
```

ql = query,  
list



```
[root@debian:~]# dpkg -L dpkg
/.
/var
/var/lib
/var/lib/dpkg
/var/lib/dpkg/updates
/var/lib/dpkg/parts
/var/lib/dpkg/info
joe ...
```

L = list

- Paketinformationen (  ):
  - `rpm -qi rpm`

```
[root@redhat:~]# rpm -qi rpm
Name           : rpm                      Relocations: (not relocatable)
Version        : 4.8.0                Vendor: opensUSE
Release        : 27.1                 Build Date: Do 17 Feb 2011 19:09:21 CET
Install Date: Di 15 Mär 2011 11:01:20 CET Build Host: build24
Group          : System/Packages      Source RPM: rpm-4.8.0-27.1.src.rpm
Size           : 3674658               License: GPLv2+
Signature      : RSA/8, Do 17 Feb 2011 19:09:42 CET, Key ID b88b2fd43dbdc284
Packager       : http://bugs.opensuse.org
Summary        : The RPM Package Manager
Description    :
RPM Package Manager is the main tool for managing the software packages
of the SuSE Linux distribution.
```

RPM can be used to install and remove software packages. With rpm, it is easy to update packages. RPM keeps track of all these manipulations in a central database. This way it is possible to get an overview of all installed packages. RPM also supports database queries.

qi = query,  
info

- Paketinformationen (  ):
  - `[root@debian:~]# dpkg -s dpkg`

s = status

```
Package: dpkg
Essential: yes
Status: install ok installed
Priority: required
Section: admin
Installed-Size: 7276
Origin: debian
Maintainer: Dpkg Developers <debian-dpkg@lists.debian.org>
Architecture: i386
Version: 1.14.31
Replaces: manpages-de (<= 0.4-3), manpages-pl (<= 20051117-1)
Pre-Depends: libc6 (>= 2.7-1), coreutils (>= 5.93-1), lzma
Suggests: apt
Conflicts: apt (<< 0.7.7), aptitude (<< 0.4.7-1), dpkg-dev (<< 1.14.16)
Conffiles:
 /etc/logrotate.d/dpkg 501f8c90b83c7ea180868ca82e1e82d1
 /etc/dpkg/origins/debian 731423fa8ba067262f8ef37882d1e742
 /etc/dpkg/dpkg.cfg f4413ffb515f8f753624ae3bb365b81b
Description: Debian package management system
 This package provides the low-level infrastructure for handling the
 installation and removal of Debian software packages.
.
 For Debian package development tools, install dpkg-dev.
Homepage: http://wiki.debian.org/Teams/Dpkg
```

## Paketkonvertierung

- `alien` wandelt RPM- und DEB-Archiv ineinander um

```
[root@debian:~]# alien --to-rpm nmap_4.62-1_i386.deb
Warning: Skipping conversion of scripts in package nmap: postinst preem
Warning: Use the --scripts parameter to include the scripts.
nmap-4.62-2.i386.rpm generated
[root@debian:~]# ls -l
-rw-r--r-- 1 root root 1054002 May 13 20:55 nmap-4.62-2.i386.rpm
-rw-r--r-- 1 root root 1048536 May 13 20:54 nmap_4.62-1_i386.deb
[root@debian:~]# rpm -qip nmap-4.62-2.i386.rpm
Name           : nmap                               Relocations: (not relocatable)
Version        : 4.62                       Vendor: (none)
Release        : 2                          Build Date: Fri May 13 20:55:33 2011
Install Date: (not installed)               Build Host: hgesser.com
Group          : Converted/net              Source RPM: nmap-4.62-2.src.rpm
Size           : 3461471                     License: see /usr/share/doc/nmap/copyri
Signature      : (none)
Summary        : The Network Mapper
Description    :
Nmap is a utility for network exploration or security auditing. It
supports ping scanning (determine which hosts are up), many port
...
```

## Aufbau der Paketnamen

**nmap-4.62-1-i386.deb**

**nmap-4.62-2-i586.rpm**

Paket-  
name

Programm-  
version

Build-  
Version

Platt-  
form

Paket-  
format

### Plattformen:

i386: Intel i386  
 i486, i586, i686: ...  
 sparc: Sun SPARC  
 ppc: IBM, Apple  
 ppc64: IBM PowerPC  
 axp: DEC Alpha  
 ia64: Intel Itanium  
 x86\_64: PC, 64 Bit  
 noarch: übergreifend

```
# dpkg-architecture -L
armel
lpia
i386
ia64
alpha
amd64
armeb
arm
...
```





- `rpm` und `dpkg` werden heute nur noch selten direkt benutzt
- stattdessen Paketverwaltung mit „höheren“ Tools wie YUM und APT

- Ein Paket hat **Abhängigkeiten** (engl. **dependencies**), wenn weitere Pakete oder bestimmte Dateien zwingend installiert sein müssen, damit es funktioniert
- Diese zusätzlich benötigten Pakete / Dateien nennt man die Abhängigkeiten
- Beispiele:
  - Paket `mc` (Midnight Commander) benötigt die Bibliothek `libc.so.6`
  - Paket `openssh-askpass` benötigt Paket `openssh`

- Abhängigkeit von Paket meist (relativ) leicht lösbar:
  - Paket suchen und installieren
  - richtige Versionsnummer, Linux-Distribution (und -Version), Plattform beachten!
- Abhängigkeit von Datei (i.d.R. Bibliothek) oft schwieriger zu beheben
  - „Welches Paket enthält die Datei `x.y.so.1`?“

- Installation bei **nicht erfüllten Abhängigkeiten** lässt sich trotzdem erzwingen:



```
[root@redhat:~]# rpm -i --nodeps paket.rpm
```



```
[root@debian:~]# dpkg -i --force-depends paket.deb
```

- Ratsam ist das aber nicht; Software wird meist nicht (oder nicht korrekt) arbeiten

- Informationen zu Abhängigkeiten:



```
[root@redhat:~]# rpm -qp --requires
mtools-4.0.15-5.6.1.i586.rpm
/bin/sh
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rpmlib(CompressedFileNames) <= 3.0.4-1
libc.so.6
libc.so.6(GLIBC_2.0)
rpmlib(PayloadIsLzma) <= 4.4.6-1
...
```

-qp: query  
package



```
[root@debian:~]# dpkg -f nmap_4.62-1_i386.deb depends
libc6 (>= 2.7-1), libgcc1 (>= 1:4.1.1),
libpcap0.8 (>= 0.9.3-1), libpcrc3 (>= 7.4),
libssl0.9.8 (>= 0.9.8f-5), libstdc++6 (>= 4.2.1)
```

-f: field

**Achtung Syntax:** Nach -f erst Paketname, dann Feldname depends

- Zwei Pakete können miteinander in **Konflikt** stehen, d. h.: Es darf nur eines von beiden installiert sein.
- Konflikt tritt meist auf, wenn man versucht, ein neues Paket zu installieren, und bereits ein damit in Konflikt stehendes installiert ist.
- Beispiele:
  - zwei Mail-Server, zwei DHCP-Server (feste Ports)

- Installation bei **Konflikten** lässt sich trotzdem erzwingen:



```
[root@redhat:~]# rpm -i --force paket.rpm
```



```
[root@debian:~]# dpkg -i --force-conflicts paket.deb
```

- Auch das Ignorieren von Konflikten ist nicht ratsam.



- Repository:
  - Sammlung von Software-Paketen
  - für genau eine Distributionsversion (z. B. OpenSuse 11.4 oder Ubuntu 10.10)
  - mit Inhaltsbeschreibungen aller Pakete
  - und Abhängigkeitsinformationen zu allen Paketen
- Installations-Tool kann Abhängigkeiten damit automatisch auflösen, nötige Zusatzpakete auswählen und diese gleich mit-installieren

- Paketverwaltung mit Repos:
  - Verwaltung der Repos
    - Eintragen neuer Repos und Löschen von Repos
    - Aktualisieren vorhandener Repos
  - „eigentliche“ Paketverwaltung
    - Installation, Update, Deinstallation von Paketen über die Repo-Tools



- Debian: APT
  - Advanced Packaging Tool
  - neben Debian auch von „Debian-basierten“ Distributionen genutzt, z. B. Knoppix, Ubuntu
  - Repository-Konfiguration in `/etc/apt/sources.list`
  - Paketverwaltung mit den Befehlen `apt-get` und `apt-cache`



- Beispiel-Konfigurationsdatei `/etc/apt/sources.list`:

```
deb cdrom:[Debian GNU/Linux 4.0]/ etch contrib main
deb http://ftp.gwdg.de/pub/linux/debian/debian/ etch main non-free
deb-src http://ftp.gwdg.de/pub/linux/debian/debian/ etch main
```

- Jede Zeile besteht aus vier Teilen:

- `deb` oder `deb-src`: Handelt es sich um eine Binär- oder Source-Paketquelle?
- URI zur Quelle, z. B. mit `http://`, `ftp://` oder `cdrom:[Volume-Name]`
- Distributionsversion (`etch`, `sarge`, `squeeze` etc.; alternativ: `stable`, `unstable`, `testing`)
- einzubindende (Unter-) Repositories: `main`, `contrib`, `non-free` etc.



- Neues Repo anlegen (Web):
  - Zusätzliche Zeile in `sources.list` eintragen
  - `apt-get update` ausführen

```
server:~ # apt-get update
Hole:1 http://security.debian.org etch/updates Release.gpg [189B]
Hole:2 http://security.debian.org etch/updates Release [22,5kB]
Hole:3 http://http.us.debian.org etch Release.gpg [378B]
Hole:4 http://http.us.debian.org etch Release [58,2kB]
Hole:5 http://security.debian.org etch/updates/main Packages [91,5kB]
Hole:6 http://http.us.debian.org etch/main Packages [4223kB]
Hole:7 http://security.debian.org etch/updates/contrib Packages [14B]
Hole:8 http://security.debian.org etch/updates/non-free Packages [14B]
Hole:9 http://http.us.debian.org etch/contrib Packages [50,6kB]
Hole:10 http://http.us.debian.org etch/non-free Packages [63,9kB]
Es wurden 4510kB in 15s geholt (291kB/s)
Paketlisten werden gelesen... Fertig
```



- Neues Repo (CD/DVD) anlegen: `apt-cdrom`

```
server:~ # apt-cdrom add
Verwendeter CD-ROM-Einbindungspunkt: /media/cdrom0/
Identifizieren ... [e1947a0c703f32d960fbfce6c7961521-2]
Durchsuchen des Mediums nach Index-Dateien ...
1 Paketindizes, 0 Quellindizes, 0 Übersetzungsindizes und 0
Signaturen gefunden
Dieses Medium heißt:
»Debian GNU/Linux 6.0.1a _Squeeze_ - Official i386 xfce+lxde-CD
Binary-1 20110322-15:11«
Reading Package Indexes... Fertig
Schreiben der neuen Quellliste
Quelllisteneinträge für dieses Medium sind:
deb cdrom:[Debian GNU/Linux 6.0.1a _Squeeze_ - Official i386
xfce+lxde-CD Binary-1 20110322-15:11]/ squeeze main
Wiederholen Sie dieses Prozedere für die restlichen Disks Ihres
Satzes.
```

→ erzeugt auch neuen Eintrag in `/etc/apt/sources.list`



- Pakete suchen: `apt-cache search`

```
server:~ # apt-cache search kmenu
deskmenu - A root menu for X11 window managers
kmenuedit - menu editor for KDE
oroborus - A lightweight themeable windowmanager for X
tastymenu - replacement K-menu for KDE/Kicker
```

- Pakete installieren: `apt-get install`

```
server:~ # apt-get install lynx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  lynx-cur
The following NEW packages will be installed:
  lynx lynx-cur
0 upgraded, 2 newly installed, 0 to remove and 6 not upgraded.
Need to get 2040kB of archives.
After this operation, 4518kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```



## Abhängigkeiten

```

root@debian:/home/esser# apt-get install k3b
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut
Statusinformationen werden eingelesen... Fertig
Die folgenden zusätzlichen Pakete werden installiert:
 cdparanoia cdrdao dvd+rw-tools exiv2 genisoimage k3b-data kaboom kbase-runtime kbase-runtime-data kdelibs-bin
 kdelibs5-data kdelibs5-plugins kdoctools libao-common libao4 libasyncns0 libattica0 libclucene0ldb1 libeggdbus-1-0
 libexiv2-9 libflac++6 libgomp1 libgraphicsmagick3 libilmbase6 libiodbc2 libjack-jackd2-0 libk3b6 libk3b6-extracodecs
 libkcdcb4 libkde3support4 libkdecore5 libkdesu5 libkdeui5 libkdnssd4 libkfile4 libkhtml5 libkio5 libkjsapi4
 libkjsembed4 libkmediaplayer4 libknewstuff2-4 libknewstuff3-4 libknotifyconfig4 libkntlm4 libkparts4 libkpty4
 libkrosscore4 libktexteditor4 libkutils4 libmusicbrainz4c2a libmysqlclient16 libnepomuk4 libnepomukquery4a libopenexr6
 libphonon4 libplasma3 libpolkit-agent-1-0 libpolkit-gobject-1-0 libpolkit-qt-1-0 libpulse-mainloop-glib0 libpulse0
 libqca2 libqt4-dbus libqt4-designer libqt4-network libqt4-opengl libqt4-qt3support libqt4-script libqt4-sql libqt4-sql-
 mysql libqt4-svg libqt4-webkit libqt4-xml libqt4-xmlpatterns libreadline5 libsamplerate0 libsndfile1 libsolid4
 libsoprano4 libssh-4 libstreamanalyzer0 libstreams0 libthreadweaver4 libvirtodbc0 libvorbisfile3 libwmf0.2-7 libxcb-
 shape0 libxine1 libxine1-bin libxine1-console libxine1-ffmpeg libxine1-misc-plugins libxine1-plugins libxine1-x
 libxml2-utils libxss1 mysql-common odbcinst odbcinst1debian2 oxygen-icon-theme phonon phonon-backend-xine plasma-
 scriptengine-javascript shared-desktop-ontologies soprano-daemon ttf-dejavu ttf-dejavu-extra vcdimager virtuoso-minimal
 virtuoso-opensource-6.1-bin virtuoso-opensource-6.1-common wodim xdg-utils
Die folgenden NEUEN Pakete werden installiert:
 cdparanoia cdrdao dvd+rw-tools exiv2 genisoimage k3b k3b-data kaboom kbase-runtime kbase-runtime-data kdelibs-bin
 kdelibs5-data kdelibs5-plugins kdoctools libao-common libao4 libasyncns0 libattica0 libclucene0ldb1 libeggdbus-1-0
 libexiv2-9 libflac++6 libgomp1 libgraphicsmagick3 libilmbase6 libiodbc2 libjack-jackd2-0 libk3b6 libk3b6-extracodecs
 libkcdcb4 libkde3support4 libkdecore5 libkdesu5 libkdeui5 libkdnssd4 libkfile4 libkhtml5 libkio5 libkjsapi4
 libkjsembed4 libkmediaplayer4 libknewstuff2-4 libknewstuff3-4 libknotifyconfig4 libkntlm4 libkparts4 libkpty4
 libkrosscore4 libktexteditor4 libkutils4 libmusicbrainz4c2a libmysqlclient16 libnepomuk4 libnepomukquery4a libopenexr6
 libphonon4 libplasma3 libpolkit-agent-1-0 libpolkit-gobject-1-0 libpolkit-qt-1-0 libpulse-mainloop-glib0 libpulse0
 libqca2 libqt4-dbus libqt4-designer libqt4-network libqt4-opengl libqt4-qt3support libqt4-script libqt4-sql libqt4-sql-
 mysql libqt4-svg libqt4-webkit libqt4-xml libqt4-xmlpatterns libreadline5 libsamplerate0 libsndfile1 libsolid4
 libsoprano4 libssh-4 libstreamanalyzer0 libstreams0 libthreadweaver4 libvirtodbc0 libvorbisfile3 libwmf0.2-7 libxcb-
 shape0 libxine1 libxine1-bin libxine1-console libxine1-ffmpeg libxine1-misc-plugins libxine1-plugins libxine1-x
 libxml2-utils libxss1 mysql-common odbcinst odbcinst1debian2 oxygen-icon-theme phonon phonon-backend-xine plasma-
 scriptengine-javascript shared-desktop-ontologies soprano-daemon ttf-dejavu ttf-dejavu-extra vcdimager virtuoso-minimal
 virtuoso-opensource-6.1-bin virtuoso-opensource-6.1-common wodim xdg-utils
0 aktualisiert, 114 neu installiert, 0 zu entfernen und 27 nicht aktualisiert.
Es müssen noch 96,2 MB von 99,6 MB an Archiven heruntergeladen werden.
Nach dieser Operation werden 238 MB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren [J/n]? J

```





- Pakete entfernen

```
server:~ # apt-get remove mc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  mc
0 upgraded, 0 newly installed, 1 to remove and 6 not upgraded.
After this operation, 6402kB disk space will be freed.
Do you want to continue [Y/n]?
```

→ remove: ohne Entfernen der Konfigurationsdateien (wie `dpkg -r`)

```
server:~ # apt-get purge paketname
```

→ purge: mit Entfernen der Konfigurationsdateien (wie `dpkg -P`)



- Alle Pakete aktualisieren: `apt-get upgrade`

```
server:~ # apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  bind9-host dhcp3-client dhcp3-common dnsmutils libbind9-50
  libdns58 libisc50 libiscxx50 libiscxxfg50 liblwres50
  libmozjs1d libnss3-1d libtiff4 mysql-common xulrunner-1.9
15 upgraded, 0 newly installed, 0 to remove and 6 not
upgraded.
Need to get 9825kB of archives.
After this operation, 24.6kB of additional disk space will
be used.
Do you want to continue [Y/n]? Y
Get:1 http://update.onlinehome-server.info lenny/updates/main
dhcp3-client 3.1.1-6+lenny5 [225kB]
Get:2 http://update.onlinehome-server.info lenny/updates/main
dhcp3-common 3.1.1-6+lenny5 [291kB]
[...]
```



- Umstieg auf neue Distributionsversion:
  - Repository-Einträge in `sources.list` auf neue Version umstellen
  - Dann `apt-get dist-upgrade` ausführen
  - Dabei werden evtl. einige Pakete entfernt, für die es in der neuen Version keinen Ersatz gibt



## Übung:

- Tragen Sie die Zeile

```
deb http://ftp.debian.org/debian/ squeeze main
```

in die APT-Konfigurationsdatei ein (falls noch nicht vorhanden – nicht mit einem ähnlich aussehenden Eintrag für Updates verwechseln!)

- Aktualisieren Sie die Paketlisten
- Installieren Sie über die APT-Tools den grafischen Editor `nedit`

Hinweis zum Netzwerk: Wenn das Netzwerk nicht funktioniert, geben Sie in der Shell `su` (danach das root-Passwort), `killall dhclient` und `dhclient eth0` ein – danach sollte es gehen. Evtl. ist eine Anmeldung des Rechners im Browser (`fom.de`) nötig.

- In der „RPM-Welt“ gibt es kein einheitliches APT-ähnliches Tool für die Verwaltung von Repositories:
  - Fedora / Red Hat: YUM
  - Mandriva: URPMI
  - Suse: Zypper
- Wir betrachten hier nur YUM
- Prinzipien sind dieselben wie bei APT; aber Eintragen der Repos funktioniert anders

- Red Hat / Fedora: YUM
  - Yellow Dog Updater, Modified
    - Yellow Dog Linux ist/war eine Linux-Distribution für PowerPC-Prozessoren, z. B. die alten Apple-Macs vor der Intel-Umstellung)
    - YUM war zunächst deren Paketmanager
  - Repository-Konfiguration: einzelne Dateien in `/etc/yum.repos.d/`
  - Paketverwaltung mit dem Befehl `yum`

- Beispiel-Konfigurationsdatei für ein Repo:

```
[livna]
name=rpm.livna.org for $releasever - $basearch
#baseurl=http://rpm.livna.org/repo/$releasever/$basearch/ http://ftp-
stud.fht-esslingen.de/pub/Mirrors/rpm.livna.org/repo/$releasever/
$basearch/
mirrorlist=http://rpm.livna.org/mirrorlist
failovermethod=roundrobin
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-livna
```

- anders als bei APT: diese Dateien nicht ändern
- einfach in den Ordner `/etc/yum.repos.d/` kopieren

- Neues Repo anlegen:
  - Repo-Anbieter stellt spezielle RPM-Datei zur Verfügung, welche das Repo automatisch einrichtet
  - Installation z. B. mit
 

```
rpm -i http://server/pfad/repopaket.rpm
```

 möglich
- Repos anzeigen: `yum repolist all`

```
server:~ # yum repolist all
Loading "installonlyn" plugin
repo id          repo name          status
development      Fedora - Development disabled
development-debuginf Fedora - Development - Debug disabled
development-source Fedora - Development - Source disabled
fedora            Fedora 7 - i386    enabled
fedora-debuginfo  Fedora 7 - i386 - Debug disabled
```



- Pakete suchen: `yum search`

```
server:~ # yum search mplayer
[...]  
mplayer.i386 1.0-0.75.20070513svn.1 livna  
Matched from:  
mplayer  
MPlayer is a movie player that plays most MPEG, VOB, AVI, OGG/OGM,  
VIVO, ASF/WMA/WMV, QT/MOV/MP4, FLI, RM, [...]
```

- Pakete installieren: `yum install`

```
server:~ # yum install w3m  
Setting up Install Process  
Parsing package install arguments  
Resolving Dependencies  
--> Running transaction check  
---> Package w3m.i386 0:0.5.2-1.fc7 set to be updated  
--> Processing Dependency: libgc.so.1 for package: w3m  
--> Restarting Dependency Resolution with new changes.  
[...]
```

- Pakete entfernen

```
server:~ # yum remove glibc
```

```
[...]
```

```
Transaction Summary
```

```
=====
```

```
Install      0 Package(s)
Update       0 Package(s)
Remove      989 Package(s)
```

```
Is this ok [y/N]? n
```

```
Exiting on user command.
Complete!
```

- YUM schlägt vor, auch alle abhängigen Pakete automatisch mit zu entfernen

- Alle Pakete aktualisieren: `yum update` (vgl.: `apt-get upgrade`)
- gleicher Effekt wie bei APT
- Distributions-Upgrade:
  - RPM-Pakete mit Repo-Informationen der neuen Version einspielen, z. B.:

```
rpm -Uvh ftp://download.fedora.redhat.com/pub/fedora/linux/releases/8/Fedora/i386/os/Packages/fedora-release-8-3.noarch.rpm ftp://download.fedora.redhat.com/pub/fedora/linux/releases/8/Fedora/i386/os/Packages/fedora-release-notes-8.0.0-3.noarch.rpm
```

- dann `yum upgrade` ausführen