

Betriebssysteme

WS 2014/15

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

Foliensatz E:

- Paketverwaltung

v1.1, 2014/11/21

Software-Verwaltung

Übersicht: BS Praxis und BS Theorie

- | | |
|-----------------|----------------------------|
| Einführung | System Calls |
| Shell | Scheduler / Prioritäten |
| Dateiverwaltung | Synchronisation |
| Filter | Speicherverwaltung |
| C-Compiler | Software-Verwaltung |
| Prozesse / Jobs | Partitionen |
| Threads | Auskunft |
| Interrupts | User, Gruppen, Rechte |

Folien E

Grundlagen Paketverwaltung (1)

- 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 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 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 (2)
 - Anlegen / Aktualisieren / Entfernen von Repos (→ parallele Nutzung mehrerer Repos möglich)
 - Suche nach Paketen in allen eingetragenen Repos
 - Installation mit automatischem Auflösen von Abhängigkeiten
 - Komplet-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 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 ...
  
```


oder

```

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


- 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)
  
```

- 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 ():

```
[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
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
```

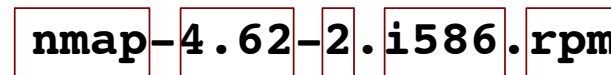
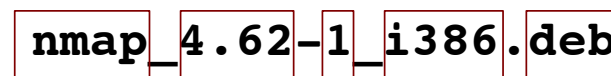
s = status

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 premm
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



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
xp: DEC Alpha
ia64: Intel Itanium
x86_64: PC, 64 Bit
noarch: übergreifend

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

Vollständige Paketliste:

```

 # rpm -qa
gpg-pubkey-3dbdc284-4be1884d
bundle-lang-gnome-en-11.4-5.13.1.noarch          -qa: query, all
translation-update-11.3-7.1.noarch
yast2-trans-stats-2.19.0-4.1.noarch
openSUSE-release-ftp-11.4-1.9.i586
apparmor-docs-2.5.1.r1445-52.55.1.i586
lsscsi-0.23-6.1.i586
libspeex1-1.1.999_1.2rc1-9.1.i586
...

 # dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name          Version          Description
-----+-----+-----+-----
ii  adduser         3.110            add and remove users and groups
ii  alien           8.72             convert and install rpm and other package
ii  apache2         2.2.9-10+lenny9 Apache HTTP Server metapackage
ii  apache2-mpm-prefork 2.2.9-10+lenny9 Apache HTTP Server - traditional non-thre
ii  apache2-utils  2.2.9-10+lenny9 utility programs for web servers
ii  apache2.2-common 2.2.9-10+lenny9 Apache HTTP Server common files
ii  apt             0.7.20.2+lenny2 Advanced front-end for dpkg
...

```


- 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`

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

- 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:

```
fpm [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

- 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)

- Informationen zu Abhängigkeiten:


```
fpm [root@redhat:~]# rpm -qp --requires          -qp: query
      mtools-4.0.15-5.6.1.i586.rpm              package
      /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
      ...
```

```
 [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), libpcre3 (>= 7.4),          -f: field
      libssl0.9.8 (>= 0.9.8f-5), libstdc++6 (>= 4.2.1)
```

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

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

```
fpm [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



- 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`

- 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



- 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.



- 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)



- 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



- 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 dnstools libbind9-50
 libdns58 libisc50 libisc50 libisc50 libisc50 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]
[...]
```



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

- 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

- 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`

- 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.  
[...]
```

- 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

- 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