

4 Vorgänge automatisch starten

Systemverwalter, die Linux-Server einrichten und verwalten wollen, sollten sich mit

- Betriebsarten,
- der Zeitsteuerung von Prozessen und
- dem hintergründigen Wirken von Dämonen gründlich vertraut machen.

Administratoren, die bisher nur mit proprietären Servern von Novell oder Microsoft gearbeitet haben, sollten sich spätestens hier mit diesen grundlegenden Linux-Konzepten vertraut machen. Unix- und Linux-Profis können getrost weiterblättern.

- Der Abschnitt 4.1 (Runlevel) beschreibt Betriebsarten zum Starten und Stoppen des Systems, für Verwaltungsarbeiten und für Mehrbenutzerbetrieb mit und ohne Netz oder Dienste.
- Zeitgesteuerte Einzelaufträge mit dem `at`-Befehl finden Sie im Abschnitt 4.2.
- Regelmäßige Vorgänge mit `cron` (Abschnitt 4.3) nehmen Systemverwaltern viele Routinearbeiten ab.
- Der Superdämon `Inetd` bzw. `Xinetd` (Abschnitt 4.4) kann im Hintergrund viele Kommunikationsdienste an straffen Zügeln lenken.

4.1 Die Runlevel von OpenSUSE

Das Mehrbenutzer-Betriebssystem Linux kennt verschiedene Betriebszustände (Runlevel) für normales Arbeiten, Wartung und Neustart.

Ein normaler Bootvorgang bringt das Linux-System in den Runlevel 5, bei dem die grafische Oberfläche mit voller Netzwerkkunterstützung aktiviert ist und mehrere Benutzer gleichzeitig mit dem System arbeiten können.

Hinweis: Gegenüber früheren Versionen von SUSE-Linux ist die Benennung der Runlevel stark verändert worden. Falls Sie Erfahrungen mit älteren Linux-Versionen haben, sollten Sie sich in `/sbin/init.d.README` mit den Änderungen vertraut machen.

Das Wechseln der Runlevel stoppt und startet Programme. So enthalten u. a. viele Konfigurationsbeschreibungen die Anweisung, die Netzwerkprogramme mit

```
init 1
init 5
```

neu zu starten. Dies wechselt zweimal den System-Zustand (Runlevel).

OpenSUSE 11.0 kennt die folgenden Runlevel:

<i>Runlevel</i>	<i>Bedeutung</i>
0	Halt
5	Single User Mode
1	Single User Mode ohne Netzwerk
2	Multi User ohne Netzwerk
3	Multi User mit Netzwerk
4	Unbenutzt
5	Multiuser mit Netzwerk und grafischer Anmeldung
6	Reboot

Tabelle 4.1: Die Runlevel von OpenSUSE

Andere Distributionen können abweichende Runlevel haben.

Mit dem Befehl

- `init 0` hält man das System an, ebenso wie mit dem Befehl `halt`.
- `init 5` wechseln Sie in den *Single User Mode*, bei dem Ihnen nur eine einzige Konsole zur Verfügung steht. Sie müssen sich nach dem Wechsel neu anmelden.
- `init 1` wechselt man in den Modus *Single User ohne Netzwerk*. Dies stoppt u. a. alle Programme, die mit dem Netzwerk zusammenhängen. Sie müssen sich nach dem Wechsel erneut anmelden.
- `init 2` wechselt das System wieder in den Modus *Multi User ohne Netzwerk* und stoppt alle Netzwerkprogramme.
- `init 3` aktiviert man den *Multi User Modus mit Netzwerk* und startet dabei alle Netzwerkprogramme neu.
- `init 5` aktiviert man den Modus *Multiuser mit Netzwerk und grafischer Anmeldung* und startet sowohl das Netzwerk als auch die grafische Oberfläche neu.
- `init 6` startet man das System neu, bewirkt also ein Reboot.

Programme, die auf einen Wechsel des Runlevels reagieren sollen, müssen im Ordner `/etc/init.d` ein Programmskript besitzen, das auf die Kommandoparameter `start` bzw. `stop` reagieren kann.

Für den im zweiten Kapitel nachinstallierten DHCPD heißt das Programmskript `dhcpd`.

Mit den Befehlen

```
/etc/init.d/dhcpd start
```

ruft man den DHCPD auf und mit

```
/etc/init.d/dhcpd stop
```

beendet man ihn wieder.

Den Aufruf vereinfachen symbolische Links, die OpenSUSE jeweils im Verzeichnis `/usr/sbin` ablegt. Ihren Namen sind jeweils die Buchstaben `rc` vorangestellt. Für den DHCPD finden Sie also in `/usr/sbin` den Link `rcdhcpd`. Da das Verzeichnis `/usr/sbin` im Suchpfad aller Benutzer liegt, können Sie den DHCPD auch ohne Pfadangabe mit

```
rcdhcpd start
```

aufrufen und mit

```
rcdhcpd stop
```

wieder beenden.

Statt per Konsole können Sie mit der grafischen Oberfläche starten und stoppen.

Sie finden im YaST-Kontrollzentrum unter *System • Runlevel-Editor • Experten-Modus* eine Auswahl der Dienste mit Startskript im Verzeichnis `/etc/init.d`.

Wenn Sie dort den Leuchtbalken auf die Zeile für den `dhcpd` führen und auf *Starten/Anhalten/Aktualisieren • Jetzt starten ...* klicken, führt YaST im Hintergrund den Befehl `dhcpd start` aus. Wenn Sie hier mit YaST arbeiten, sollten Sie für `dhcpd` die Runlevel 3 und 5 auswählen.

Das dreiteilige Startskript `dhcpd` ist einigermaßen lesbar. Der erste Teil beschreibt, in welchen Runleveln der Dämon laufen soll und welche anderen Dienste vorher gestartet sein müssen. YaST und andere Konfigurationstools werten diesen Teil aus, auch wenn er durch #-Zeichen auskommentiert ist.

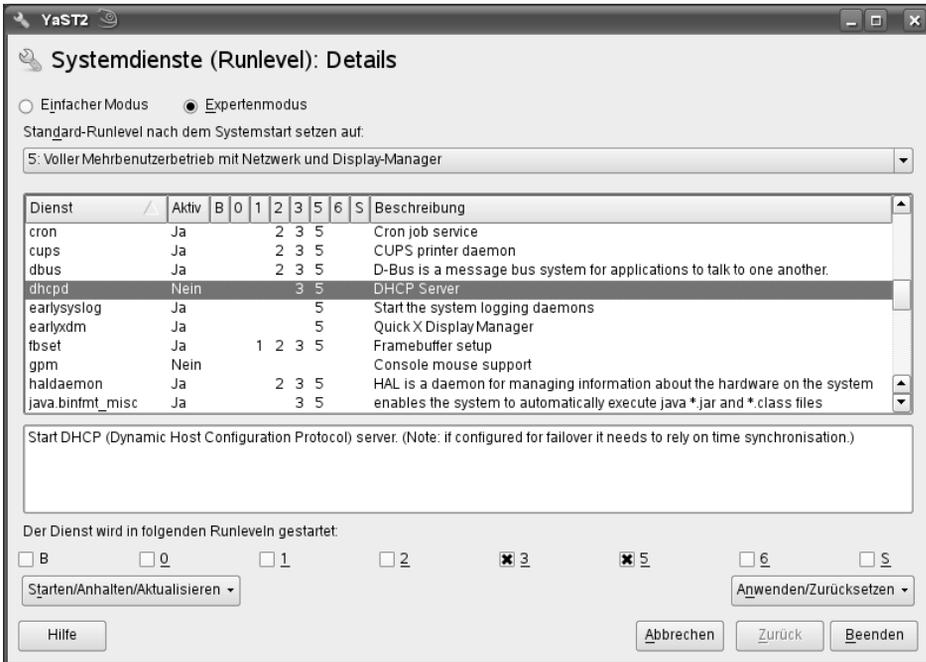


Abbildung 4.1: Runlevel-Editor

Der nächste Abschnitt nennt die Pfade zu Programm und Konfigurationsdateien und das Skript, welches das Vorhandensein einiger Dateien prüft.

Erst im dritten Teil folgen die Programmzeilen für die einzelnen Parameter, mit denen Sie das Skript aufrufen können.

Wer sich für die Firmengeschichte der Firma SUSE interessiert, dürfte sich auch für die Copyright-Zeilen am Anfang der Datei interessieren. In späteren Versionen wird hier vielleicht auch ein Hinweis auf Novell bzw. OpenSUSE auftauchen.

/etc/init.d/dhcp (Dateianfang):

```
#!/bin/sh
# Copyright (c) 1996, 1997, 1998 S. u.S.E. GmbH
# Copyright (c) 1998, 1999, 2000, 2001 SuSE GmbH
# Copyright (c) 2002, 2003 SuSE Linux AG
#
# Author: Rolf Habercker <rolf@suse.de>, 1997, 1998, 1999
#         Peter Poeml <poeml@suse.de>, 2000, 2001, 2002, 2003
#
# /etc/init.d/dhcpd
# and its symbolic link
# /usr/sbin/rcdhcpd
#
### BEGIN INIT INFO
```

```

# Provides:                                dhcpd
# Required-Start:                          $remote_fs $network
# Should-Start:                            $named $syslog $time ldap ndsd
# Required-Stop:                          $remote_fs $network
# Should-Stop:                            $named $syslog ldap ndsd
# Default-Start:                          3 5
# Default-Stop:                            0 1 2 6
# Short-DeSkription:                      DHCP Server
# DeSkription:                            Start DHCP (Dynamic Host
#                                           Configuration Protocol)
#                                           server. (Note: if configured
#                                           for failover it
#                                           needs to rely on time
#                                           synchronisation.)
#### END INIT INFO

if [ -s /etc/sysconfig/dhcpd ]; then

    . /etc/sysconfig/dhcpd

else

    # pre 8.0

    # Source SuSE config
    . /etc/rc.config

    test -s /etc/rc.config.d/dhcpd.rc.config && \
        . /etc/rc.config.d/dhcpd.rc.config

    # Determine the base and follow a runlevel link name.
    base=${0##*/}
    link=${base#*[SK][0-9][0-9]}

    # Force execution if not called by a runlevel
    # directory.
    test $link = $base && START_DHCPD=yes
    test "$START_DHCPD" = yes || exit 0

fi

```

Am Ende des Ausschnitts erkennen Sie, wie das Skript die Sysconfig-Variablen `START_DHCPD` auswertet. Hat diese nicht den Wert `yes`, beendet sich das Skript, wenn es über einen Runlevel-Wechsel aufgerufen wurde.

Das Skript kennt außer `start` und `stop` u. a. die Parameter `reload`, `restart` und `status`. Das Kommando `restart` stoppt den `dhcpd` und startet ihn nach 3 Sekunden

wieder, `reload` liest die Konfiguration erneut und mit `status` testet das Programm, ob der `dhcpd` läuft.

Im ersten Teil prüft das Programm, ob es direkt von der Konsole aus gestartet wurde oder über einen Wechsel der Runlevel. Bei einem Start über Runlevel beachtet es die Variable `START_DHCPD` aus der Konfigurationsdatei `/etc/sysconfig/dhcpd`. Nur wenn diese den Wert `yes` hat, startet das Programm.

Jetzt fehlt noch die Kopplung an den Wechsel der Runlevel. Dazu gibt es unterhalb von `/etc/init.d` für jeden Runlevel ein Verzeichnis, also

- `/etc/init.d/rc0.d,`
- `/etc/init.d/rc1.d,`
- `/etc/init.d/rc2.d,`
- `/etc/init.d/rc3.d,`
- `/etc/init.d/rc4.d,`
- `/etc/init.d/rc5.d,`
- `/etc/init.d/rc6.d,`
- `/etc/init.d/rcS.d.`

In diesen Ordnern befinden sich ggf. Verweise (Softlinks) auf die Start-/Stopp-Dateien im Ordner `/etc/init.d`. Für den `dhcpd` sind dies z. B. die Links

- `S14dhcpd` und
- `K08dhcpd`.

Der Buchstabe `S` steht hier für *Start*, der Buchstabe `K` für *Kill* (Beenden). Beim Wechsel in den Runlevel 5 ruft das Linux-System alle Links im Verzeichnis `rc5.d`, die mit einem `S` beginnen, mit dem Parameter `start` auf. Die Zahl gibt eine Reihenfolge an; je höher die Zahl, desto später startet das zugehörige Programm.

Beim Verlassen eines Runlevels wertet Linux die Links aus, die mit einem `K` beginnen. Das zugehörige Programmskript startet dann mit dem Parameter `stop`.

Die Distribution OpenSUSE installiert die Startskripte und Links der Programme automatisch. Bei Programmen, die vor ihrem Start noch konfiguriert werden müssen, stehen in der Konfigurationsdatei in `/etc/sysconfig/` die entsprechenden Startschalter (z. B. `DHCP_START`) noch auf `no`.

Im Kapitel 12 (Firewalling und Masquerading) finden Sie ein eigenes Programmskript `/etc/init.d/maske`. Wenn dieses im Runlevel 3 aktiv sein soll, müssen Sie in `/etc/init.d/rc3.d` folgende Links anlegen:

```
ln -s /etc/init.d/maske /etc/init.d/rc3.d/S40maske
ln -s /etc/init.d/maske /etc/init.d/rc3.d/K40maske
```

Damit startet das Programm beim Wechsel in den Runlevel 3 und stoppt beim Verlassen des Runlevels 3.

Ein Muster für eigene Startprogramme finden Sie in der Datei `/etc/init.d/skeleton`.

`/etc/init.d/skeleton` (Auszug, Dateianfang):

```
#!/bin/sh
#
#   Template SUSE system startup Skript for example
#   service/daemon FOO
#   Copyright (C) 1995--2005 Kurt Garloff, SUSE / Novell
#   Inc.
#
#
#
...
#
#### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Should-Start:     $time ypbind sendmail
# Required-Stop:     $syslog $remote_fs
# Should-Stop:      $time ypbind sendmail
# Default-Start:    3 5
# Default-Stop:     0 1 2 6
# Short-DeSkription: FOO XYZ daemon providing ZYX
# DeSkription:      Start FOO to allow XY and provide YZ
#                   continued on second line by '#<TAB>'
#                   should contain enough info for the runlevel editor
#                   to give admin some idea what this service does and
#                   what it's needed for ...
#                   (The Short-DeSkription should already be a good hint.)
#### END INIT INFO
#
#
...
# * The services specified in the stop tags
#   (Required-Stop/Should-Stop)
#   specify which services need to be still running when this
#   service is shut down.
#   Often the entries there are just copies or a subset
#   from the respective start tag.
# * Should-Start/Stop are now part of LSB as of 2.0,
#   formerly SUSE/Unitedlinux used X-UnitedLinux-Should-
#   Start/-Stop.
#   insserv does support both variants.
```

```
# * X-UnitedLinux-Default-Enabled: yes/no is used at
# installation time
# (%fillup_and_insserv macro in %post of many RPMs) to
# specify whether
# a startup Skript should default to be enabled after
# installation. It's not used by insserv.
#
# Note on runlevels:
# 0 - halt/poweroff
# 6 - reboot
# 1 - single user
# 2 - multiuser without network exported
# 3 - multiuser w/ network (text mode)
# 5 - multiuser w/ network and X11 (xdm)
```

Diese Datei können Sie an Ihre Bedürfnisse anpassen. Relativ neu in dieser Datei ist der Block `INIT INFO`. Hier geben Sie an, in welchen Runleveln Ihr Programm aktiv sein soll und welche anderen Dienste zuvor bereits gestartet bzw. gestoppt sein müssen.

Mit diesen Informationen kann das Programm `insserv` im Verzeichnis `/etc/init.d/` automatisch die passenden symbolischen Links für die Runlevel anlegen. Über den Wechsel der Runlevel startet das System Programme, die ständig aktiv sein sollen. Einmalig oder wiederholt startet man Programme automatisch mit `at` und `cron`:

- Mit `at` startet man ein Programm einmalig zu einem bestimmten Zeitpunkt. Eine typische Anwendung sind Wartungsarbeiten zu einer Zeit, an denen normalerweise keine Benutzer angemeldet sind.
- Programme für regelmäßige Wartungsarbeiten und statistische Auswertungen ruft man zu einem bestimmten Zeitpunkt mit `cron` auf.
- In den nächsten beiden Abschnitten erfahren Sie mehr über die Zeitsteuerung.

4.2 Zeitgesteuerte Einzel-Aufträge

Mit dem Befehl `at` können Berechtigte zu einem bestimmten Zeitpunkt Programme ausführen, zum Beispiel ein zeitaufwendiges Programm nachts starten.

Tipp: Mit `at` kann man nur Programme starten, für deren Ausführung man auch die notwendigen Rechte besitzt. Für das Beispiel sollte man als `root` angemeldet sein, da normale Benutzer mit `find` nicht in allen Verzeichnissen suchen dürfen.

Um z. B. alle Dateien zu finden, die keinem Benutzer gehören, kann man den `find`-Befehl in der folgenden Form einsetzen:

```
find / -nouser
```

Der Suchvorgang ist recht zeitaufwendig, da `find` dazu alle Dateien untersuchen muss. Dateien ohne Benutzer entstehen, wenn man Benutzer löscht, die Dateien außerhalb ihrer Home-Verzeichnisse abgelegt haben. Da die Suche in größeren Systemen recht lange dauern kann, sollte man diese Suche auf einen ruhigen Zeitpunkt, z. B. 22:00 Uhr, verschieben. Dazu gibt man ein:

```
at 22:00
```

Am veränderten Eingabezeichen gibt man den eigentlichen Befehl ein

```
at> find / -nouser
```

und schließt die Eingabe dann mit `Strg_D` ab.

```
boss:~ # at 22:00
warning: commands will be executed using /bin/sh
at> find / -nouser
at> <EOT>
job 1 at 2008-07-23 22:00
boss:~ #
```

Die Zeitpunkte für die Ausführung kann man auf verschiedene Arten angeben, wie hier im Beispiel über `HH:MM`, aber auch mit `now +2 hours`. Damit würde das Programm in zwei Stunden starten. Statt `hours` sind auch die Angaben `minutes`, `days` und `weeks` und absolute Zeitangaben wie `teatime` (16:00 Uhr) und `midnight` möglich.

Tipp: Sollten Sie nicht die beschriebene Rückmeldung erhalten, sondern eine Fehlermeldung, dann läuft der `at`-Dämon nicht und Sie müssen ihn erst starten, z. B. über den Runlevel-Editor.

Unerledigte Aufträge zeigt `atq` an:

```
boss:~ # atq
1      2008-07-23 22:00  a      root
```

Anhand der Jobnummer eines Auftrages (hier `job 1`) kann man diesen wieder löschen:

```
boss:~ # atrm 1
```

Der `at`-Dämon gibt Daten statt auf den Bildschirm in eine Mail an den Auftraggeber aus.

4.3 Regelmäßige Vorgänge mit cron

Für regelmäßige Vorgänge ist `cron` ein besseres Werkzeug als `at`. Für solche Vorgänge listet man in einer sogenannten *crontab*-Tabelle die Vorgänge und die Zeiten auf, an denen man diese ausführen will.

In der Grundeinstellung dürfen alle Benutzer, die nicht in der Datei `/etc/cron.deny` verzeichnet sind, eine `crontab` anlegen. In der Grundinstallation sind hier nur `ghost` und `guest` eingetragen.

Eine derartige `crontab`-Tabelle könnte folgendermaßen aussehen:

```
# roots crontab
#
# min hour day month dayofweek (1=Mo,7=Su) command
15 22 * * * /usr/bin/find / -nouser
```

So startet der Suchbefehl jeden Tag um 22:15 Uhr. Ein Stern steht als Jokerzeichen für alle Zeiten. Der Suchbefehl startet also an jedem Tag, in jedem Monat und an jedem Wochentag um 22:15 Uhr.

Eingeben kann man diese Tabelle als Benutzer `root` durch:

```
crontab -e
```

Die Möglichkeiten der Zeitangabe sind recht vielfältig. Mit z. B.

```
# roots crontab
#
# min hour day month dayofweek (1=Mo,7=Su) command
15 22 * * 1-5 /usr/bin/find / -nouser
```

würde die Suche nur an Werktagen ablaufen.

Zu den Anwendungen, die Sie regelmäßig per `cron` ausführen sollten, gehört die Datensicherung – das Backup. Hinweise hierzu finden Sie oben im Kapitel 2.

Neben diesen zeitgesteuerten Aufträgen, die an einen Benutzer gebunden sind, kennt `cron` auch Aufträge die an einen allgemeinen Zeitpunkt gebunden sind:

- stündlich (`hourly`),
- täglich (`daily`),
- wöchentlich (`weekly`),
- monatlich (`monthly`).

Für derartige Aufträge besitzt `cron` Verzeichnisse unterhalb von `/etc`, z. B. `/etc/cron.daily` für die täglichen Aufträge. Alle Programme die sich in einem dieser Verzeichnisse befinden, führt `cron` immer dann durch, wenn der entsprechende Zeitpunkt gekommen ist. Außer bei `cron.hourly` ist das immer um Mitternacht.

4.4 Der Super-Dämon inetd für Internetdienste

Für viele Internetdienste, wie POP, SMTP, FTP und Telnet findet man weder ein Startskript in `/etc/init.d/` noch einen zeitgesteuerten Aufruf.

Das spart Ressourcen, da die zugehörigen Programme erst bei Bedarf starten. Für das Starten ist der Super-Dämon `inetd` zuständig. OpenSUSE installiert dazu den `xinetd`, eine neuere Version des `inetd`; das `x` im Namen steht für *extended* (erweitert).

Der `xinetd` ist flexibler konfigurierbar und auch stärker auf Sicherheit ausgerichtet als sein Vorgänger.

Konfigurieren können Sie `xinetd` in der recht kurzen und übersichtlichen Datei `/etc/xinetd.conf`.

`/etc/xinetd.conf`:

```
#
# xinetd.conf
#
# Copyright (c) 1998-2001 SuSE GmbH Nuernberg, Germany.
# Copyright (c) 2002 SuSE Linux AG, Nuernberg, Germany.
#

defaults
{
    log_type           = FILE /var/log/xinetd.log
    log_on_success     = HOST EXIT DURATION
    log_on_failure     = HOST ATTEMPT
#    only_from         = localhost
    instances          = 30
    cps                = 50 10

#
# The specification of an interface is interesting, if we are
# on a firewall.
# For example, if you only want to provide services from an
# internal network interface, you may specify your internal
# interfaces IP-Address.
#
#    interface        = 127.0.0.1
#
}

includedir /etc/xinetd.d
```

Mit dem `#`-Zeichen beginnen wie üblich die Kommentarzeilen. In dieser Konfigurationsdatei finden Sie nur allgemeinere Einstellungen, z. B. den Namen der Log-Datei und welche Informationen darin aufgenommen werden sollen. Sie können hier auch

regeln, dass Dienste überhaupt nur für ein bestimmtes Interface, also z. B. nur für eth0, starten können.

Am wichtigsten ist die letzte Zeile, welche die Dateien im Verzeichnis `/etc/xinetd.d` in die Konfiguration aufnimmt. Wenn Sie wie im Kapitel 3 beschrieben das Paket `qpopper` installiert haben, finden Sie in diesem Verzeichnis jetzt eine Datei `qpopper`.

Mit dieser Datei legen Sie fest, was der `xinetd` bei Anfragen auf dem zugehörigen Port 110 machen soll.

```
/etc/xinetd.d/qpopper
#
# qpopper - pop3 mail daemon
service pop3
{
    socket_type      = stream
    protocol        = tcp
    wait            = no
    user            = root
    server          = /usr/sbin/popper
    server_args     = -s
    flags           = IPv4
}
```

Sie finden hier die Angabe des Serverprogrammes (`server`), eventuell mit zusätzlichen Startparametern (`server_args`), sowie Angaben zum Protokoll (`protocol`) und dem Benutzernamen (`user`) für diesen Dienst. Jeder Dienst arbeitet mit den Rechten eines bestimmten Benutzers (hier `root`), so als ob dieser Benutzer das Programm selbst gestartet hätte.

Wenn Sie in der Konfigurationsdatei für einen Dienst die Zeile

```
disable          = yes
```

finden, ist dieser Dienst deaktiviert und der `xinetd` wird ihn nicht starten.

Sie brauchen die Konfigurationsdateien nicht direkt zu bearbeiten. Einfacher geht das über das YaST-Kontrollzentrum und die Funktion *Netzwerkdienste • Netzwerkdienste (xinetd)*. Sie sehen dort eine Liste aller vorbereiteten Dienste und deren Status.

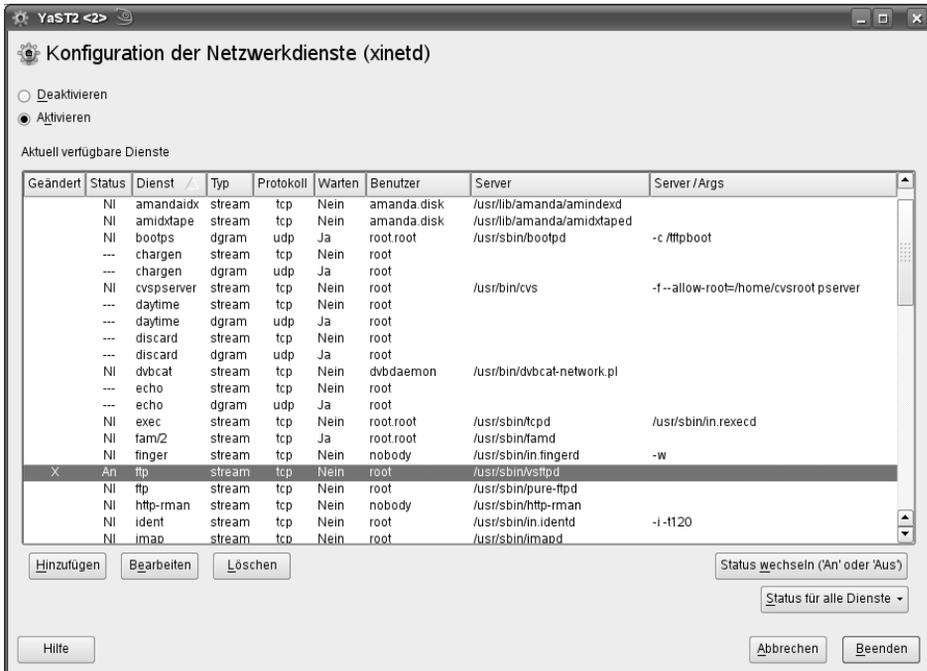


Abbildung 4.2: Konfiguration der Netzwerkdienste

Im Auslieferungszustand ist keiner der Dienste aktiviert. OpenSUSE aktiviert den `xinetd` daher auch nicht. Sie müssen also im Zweifelsfall zuerst auf *Aktivieren* gehen, um den `xinetd` überhaupt zu starten.

Dazu steuern Sie den Leuchtbalken auf den gewünschten Dienst und aktivieren (oder deaktivieren) diesen Dienst mit *Status wechseln*. Da YaST die Änderungen erst nach dem Klick auf *Weiter* in die Konfigurationsdateien übernimmt, zeigt es in der ersten Spalte der Zeile noch an, dass es sich um eine geänderte Einstellung handelt.

Mit einem Klick auf *Bearbeiten* können Sie auch die gesamte Konfigurationsdatei für einen Dienst verändern.

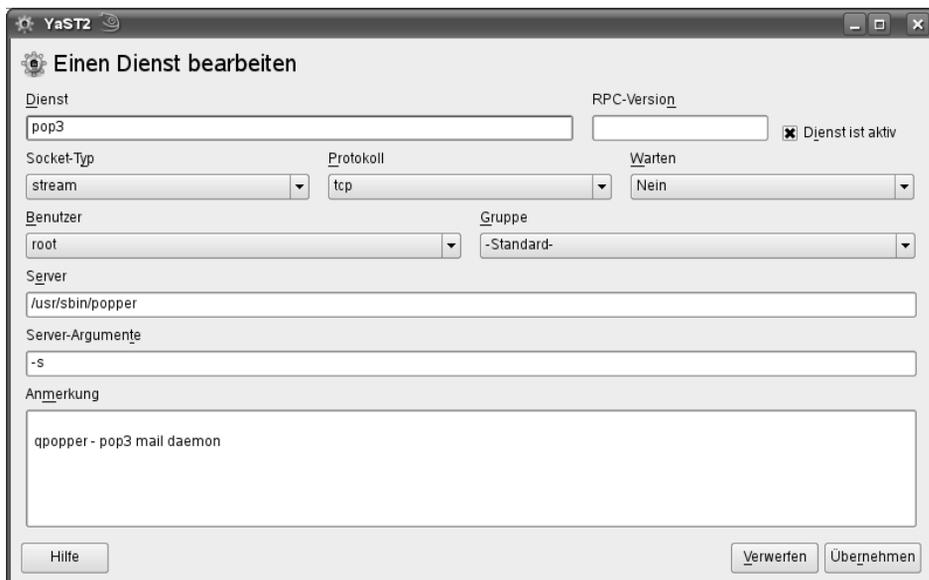


Abbildung 4.3: Einen Dienst bearbeiten

Sie sehen hier die gleichen Informationen wie in der Konfigurationsdatei. Da das Formular die Gefahr von Fehleingaben verringert, hilft es beim Konfigurieren.

YaST unterscheidet drei Dienstzustände. Am häufigsten finden Sie deaktiviert (---), am wichtigsten ist wohl aktiviert (*An*). Weiter unten in der Liste stehen die nicht installierten Dienste (*NI*), für die OpenSUSE die Konfiguration vorbereitet hat, wobei das Paket mit dem Serverdienst aber noch fehlt.

Wenn Sie dieses Buch bis zum Ende durchgearbeitet haben, werden sich in Ihrer Liste mehr aktivierte Dienste finden als jetzt. Da jeder aktivierte Dienst ein zusätzliches Risiko bedeutet, sollten Sie stets nur Dienste aktivieren, die Sie auch benötigen – insbesondere, wenn Sie Ihren Rechner direkt mit dem Internet verbinden.