

# Linux – Prinzipien und Programmierung

Dr. Klaus Höppner

Hochschule Darmstadt – Wintersemester 2011/2012

1/17

## Dienste

2/17

## Aktionen in den einzelnen Runlevels

Welche Dienste beim Betreten eines Runlevels gestartet (oder beim Wechseln aus einem anderen Runlevel gestoppt) werden, wird in Linux über verschiedene Skripten und Links gesteuert, die sich unterhalb des Verzeichnisses `/etc/rc.d` befinden:

`/etc/rc.d/init.d` Start-/Stop-Skripte (oft auch: Init-Skript) der einzelnen Dienste,

`/etc/rc.d/rc?.d` Start-/Stop-Skripte für den betreffenden Runlevel, wobei diese jeweils symbolische Links auf das entsprechende Skript in `/etc/rc.d/init.d` sind.

3/17

## Anatomie eines Start-/Stop-Skriptes

Die Start-/Stop-Skripte müssen zwingend zwei Kommandozeilenparameter zum Starten bzw. Beenden eines Dienstes verstehen:

```
/etc/rc.d/init.d/myscript start oder
/etc/rc.d/init.d/myscript stop
```

Im Allgemeinen sind noch andere Parameter zulässig:

- `status` zur Ausgabe des Status (running oder stopped),
- `restart` führt stop und start hintereinander aus,
- `condrestart` falls der Dienst läuft, wird er beendet und neu gestartet,
- `reload` weist einen laufenden Dienst an, seine Konfigurationsdateien neu einzulesen.

4/17

## Anatomie eines Start-/Stop-Skriptes (Forts.)

Was soll nun das Start-/Stop-Skript machen:

- Beim Start soll natürlich ein Dienstprozess gestartet werden.  
Hierbei wird erwartet, dass der Dienstprozess seine Prozess-ID in die Datei `/var/run/dienst.pid` schreibt. Außerdem legt das Start-Skript eine Datei `/var/lock/subsys/skriptname` an.
- Beim Stoppen wird der Prozess mit der in `/var/run/dienst.pid` stehenden Prozess-ID gestoppt, anschließend die Datei unter `/var/lock/subsys` gelöscht.
- In beiden Fällen wird auf dem Bildschirm ausgegeben, ob das Starten bzw. Stoppen des Dienstes erfolgreich war.

Für die häufig benötigten Aktionen in den Skripten gibt es vordefinierte Shell-Funktionen in der Datei `/etc/rc.d/init.d/functions`, die in den Skripten daher zu Beginn inkludiert (per Punkt-Befehl) wird.

5/17

## Ablauf beim Wechseln in einen Runlevel

In dem Verzeichnis `/etc/rc.d/rc?d` befinden sich Links auf die Start-/Stop-Skripte, die jeweils die Form `SddSkriptname` bzw. `KddSkriptname` haben, d. h diese beginnen mit S oder K und einer zweistelligen Zahl.

Beim Wechseln in diesen Runlevel werden nun zunächst die mit K beginnenden Links mit dem Parameter stop aufgerufen (also K=kill), wenn es einen entsprechenden Link unter `/var/lock/subsys` gibt, danach die mit S beginnenden Links mit dem Parameter start aufgerufen (also S=start). Hierbei werden die Stop- bzw. Start-Befehle in aufsteigender Reihenfolge der zweistelligen Zahlen ausgeführt, also lautet die Reihenfolge z. B.

```
K01abc, K85def, S25ghi, S95jkl
```

6/17

## Beispieldienst

Das folgende C-Programm schreibt als Hintergrundprozess alle 10 Sekunden eine Meldung mit dem Inhalt „ping“ über den syslog-Dämon in die Liste der Systemmeldungen:

```
#include <unistd.h>
#include <syslog.h>
#include <stdio.h>

int main() {
    int i = fork();
    if (i<0) return(1);
    if (i>0) return(0);

    FILE* pid=fopen("/var/run/log.pid","w");
    fprintf(pid,"%d\n",getpid());
    fclose(pid);
    openlog("ping",LOG_ODELAY,LOG_LOCAL0);
    while (1) {
        syslog(LOG_INFO,"%s","ping");
        sleep(10);
    }
}
```

7/17

## Das Start-/Stop-Skript

Nun legen wir das Start-/Stop-Skript als `/etc/init.d/log` an:

```
#!/bin/bash
#
# log          service to send pings to syslog
#
# chkconfig: 345 99 01
# description: pings to syslog as test service
#
# pidfile: /var/run/log.pid

. /etc/init.d/functions

RETVAL=0
prog=log
progdir=/root
CMD=$progdir/$prog
lockfile=/var/lock/subsys/$prog
```

8/17

## Das Start-/Stop-Skript (Forts.)

```
start() {
    [ "$EUID" != 0 ] && exit 4
    [ -x "$CMD" ] || exit 1

    echo -n "Starting $prog: "
    daemon $CMD
    RETVAL=$?
    [ $RETVAL -eq 0 ] && success || failure
    echo
    [ $RETVAL -eq 0 ] && touch $lockfile
    return $RETVAL
}
```

9/17

## Das Start-/Stop-Skript (Forts.)

```

stop() {
    [ "$EUID" != 0 ] && exit 4

    echo -n "Stopping $prog: "
    if [ -n "$(pidfileofproc $prog)" ]; then
        killproc $prog
    else
        failure "Stopping $prog"
    fi
    RETVAL=$?

    [ $RETVAL -eq 0 ] && rm -f $lockfile
    echo
    return $RETVAL
}

```

10/17

## Das Start-/Stop-Skript (Forts.)

```

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    status)
        status $prog
        ;;

```

11/17

## Das Start-/Stop-Skript (Forts.)

```

condrestart)
    if status $prog >/dev/null; then
        stop
        start
    fi
    ;;
reload)
    exit 4
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart|condrestart|re
    exit 2
esac

```

12/17

## Arbeiten mit Start-/Stop-Skripten

Im Prinzip kann unser Start-/Stop-Skripte von Hand gestartet werden:

```
/etc/init.d/log start oder  
/etc/init.d/log stop oder  
/etc/init.d/log status
```

Stattdessen kann aber (auf Redhat-Systemem) das Programm `service` genutzt werden, das sich in `/usr/sbin` befindet:  
`service log start` usw.

Nebenbei unterstützt `service` noch den Aufruf  
`service --status-all`

13/17

## Verwalten der Start-/Stop-Skripte

Zu einem ordnungsgemäß konfigurierten Systemdienst gehören neben dem eigentlichen Init-Skript die entsprechenden Links in den einzelnen Runlevel-Verzeichnissen, so dass der Dienst beim Wechsel in den Runlevel entweder gestartet oder beendet wird.

Soll unser Dienst `log` in den Runlevels 3, 4 und 5 laufen, so müssen also folgende Links existieren:

```
/etc/rc.d/rc0.d/K01log (Halt)  
/etc/rc.d/rc1.d/K01log  
/etc/rc.d/rc2.d/K01log  
/etc/rc.d/rc3.d/S99log  
/etc/rc.d/rc4.d/S99log  
/etc/rc.d/rc5.d/S99log  
/etc/rc.d/rc6.d/K01log (Reboot)
```

14/17

## chkconfig

Auf RedHat-Systemen existiert (übernommen von IRIX) das Programm `chkconfig`, mit dem diese Links in den Runlevel-Verzeichnissen verwaltet werden können. Dabei wird der Kommentar

```
# chkconfig: 345 99 01
```

im Init-Skript ausgewertet, die die Runlevel angibt, in denen der Dienst laufen soll, sowie die zweistellige Zahl für die Start- bzw. Stop-Links angibt.

15/17

## Benutzen von chkconfig

- `chkconfig log` prüft, ob der Dienst `log` korrekt konfiguriert ist,
- `chkconfig --add log` erzeugt dabei die Links für den Dienst `log`,
- `chkconfig --list log` listet auf, in welchen Runlevels der Dienst läuft.
- `chkconfig --del log` entfernt alle Start-/Stop-Links zum Dienst.

Die Einstellung für einzelne Runlevel kann mit `chkconfig --levels {level} log {on|off}` geändert werden.

16/17

## Beispiele für Dienste

- `ntpd`, Dämon zum laufenden Synchronisieren mit Zeitserver über das *Network Time Protocol*, optional auch als Server für andere Rechner.
- `httpd`, Webserver (i. A. Apache).
- `nfs`, zum Exportieren von Verzeichnissen an andere Rechner über NFS.
- `sshd`, zum Zulassen von Verbindungen über *Secure Shell*.
- `crond`, Dämon zum zeitgesteuerten Ausführen von Aufgaben.

17/17