

3.4.F Grundkonfiguration/Systemstart – Ergänzungen und Bilder

3.4.F.1 Beispiel für ein Init-Skript

Hier ist ein Beispiel für ein Init-Skript:

```

1  #!/bin/sh
2  # Beispiel fuer ein Init-Skript
3  if [ ! -f /usr/bin/top ]
4  then
5      exit 0
6  fi
7  case "$1" in
8      (start)
9          echo -n "starte_top_..._"
10         /usr/bin/top -b > /dev/tty9 2>&1 &
11         jobs -p %1 > /var/run/top.pid
12         echo "fertig";;
13     (stop)
14         echo "stoppe_top_..."
15         /bin/kill -TERM $(cat /var/run/top.pid)
16         echo "fertig";;
17     (reload)
18         echo "neue_Konfig._fuer_top_..."
19         /bin/kill -HUP $(cat /var/run/top.pid)
20         echo "fertig";;
21     (restart)
22         echo "stoppe_top_..."
23         /bin/kill -TERM $(cat /var/run/top.pid)
24         echo -n "fertig;_starte_top_neu_..._"
25         /usr/bin/top -b > /dev/tty9 2>&1 &
26         jobs -p %1 > /var/run/top.pid
27         echo "fertig";;
28     (*)
29         echo "usage:_$0_(start|stop|reload|restart)"
30         exit 1;;
31 esac
32 exit 0

```

3.4.F.2 Warum Upstart?

Die Methode der Init-Skripten ist eine bewährte und extrem zuverlässige Methode des Systemstarts. Für Serversysteme hat sie kaum Nachteile, auf Desktop- oder Embedded-Systemen stört jedoch ihre lange Bootdauer. Auch ihre Zusammenarbeit mit automatischer Hardware-Erkennung bei Hotplug-fähiger Hardware war nicht optimal. Daher wurde das upstart-System entwickelt, bei dem Prozesse erst bei Bedarf durch das Eintreffen besonderer Ereignisse gestartet werden und nicht nur dann, wenn der Runlevel geändert oder neu eingelesen wird. upstart soll vollständig kompatibel zu den traditionellen Methoden des Systemstarts sein.

3.4.F.3 Konfiguration bei Upstart

Bei upstart gibt es für jeden Service eine Konfigurationsdatei unter dem Pfadnamen /etc/init.d/xyz.conf, wobei xyz den Namen des Service angibt. Hier ein Beispiel:

```

1  start on runlevel [23]

```

```
2 | stop on runlevel [!23]
3 | respawn
4 | exec /sbin/getty -8 tty6
```

Die Zeilen mit `start` und `stop` geben die Abhängigkeiten an, das Wort `respawn` gibt an, dass der Prozess nach seinem Ende neu gestartet werden soll, und die `exec`-Zeile ist die Befehlszeile. Man erkennt die Ähnlichkeit zum Start mit `inittab`, aber ohne BSD-Init-Skripte. Aber hier kann man noch weitere Abhängigkeiten angeben, z.B.:

```
1 | start on      (stopped tty5 and starting tty3) \
2 |      until (stopped tty7 or stopped tty8)
```

Außerdem kann man Abschnitte für Shell-Skripte festlegen, die vor dem Start, beim Start und nach dem Start ausgeführt werden sollen:

```
1 | pre-start script
2 | ...
3 | end script
4 | script
5 | ...
6 | end script
7 | post-start script
8 | ...
9 | end script
```

In der Datei `rc-sysinit.conf` befindet sich die übergeordnete Konfiguration für `upstart`, unter anderem der Standard-Runlevel in der Variablen `DEFAULT_RUNLEVEL`.

3.4.F.4 Werkzeuge für Upstart

Standardwerkzeug für Upstart ist das Programm `initctl`. Folgende Befehle sind wichtig:

- `initctl list` – Auflisten laufender Dienste
- `initctl start sane` – SANE starten
- `initctl stop sane` – SANE stoppen
- `initctl restart sane` – SANE neu starten
- `initctl status sane` – Status von SANE anzeigen
- `initctl reload-configuration` – Inhalt von `/etc/init.d/*.conf` neu einlesen