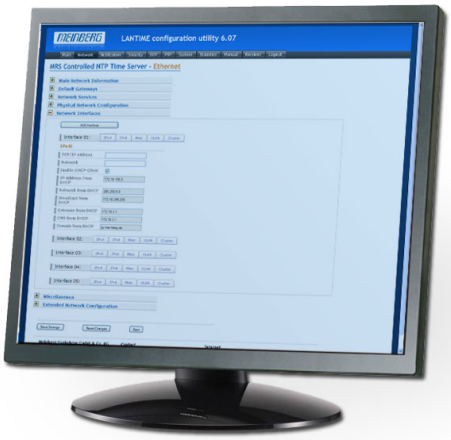

HANDBUCH

LANTIME OS Version 6

V6.24 Command Line Interface Reference

27. Oktober 2017

Meinberg Funkuhren GmbH & Co. KG

Inhaltsverzeichnis

1 Einleitung **1**

2 Verwendung des CLI **2**

3 Befehlsreferenz **3**

3.1 Konfigurationsverwaltung . . . 4
3.1.1 lsconfig - Anzeige von gespeicherten Configsets . . . 4
3.1.2 rmconfig - Löschen eines Configsets . . . 5
3.1.3 diffconfig - Ungespeicherte Änderungen anzeigen . . . 6
3.1.4 checkconfig - Prüfung auf ungespeicherte Konfigurationsänderungen . . . 6
3.1.5 saveconfig - Konfigurationsänderungen speichern . . . 7
3.1.6 loadconfig - Configset laden . . . 8
3.2 Datei Verwaltung . . . 9
3.2.1 pwd - Arbeitsverzeichnis anzeigen . . . 9
3.2.2 cd - Arbeitsverzeichnis wechseln . . . 9
3.2.3 ls - Verzeichnisinhalt anzeigen . . . 10
3.2.4 cp - Dateien und/oder Verzeichnisse kopieren . . . 11
3.2.5 mv - Dateien und/oder Verzeichnisse verschieben bzw. umbenennen . . . 12
3.2.6 rm - Dateien und/oder Verzeichnisse löschen . . . 13
3.2.7 cat - Datei anzeigen . . . 14
3.3 Firmware Verwaltung . . . 15
3.3.1 fwlist - Installierte Firmware Images anzeigen . . . 15
3.3.2 fwselect - Aktives Firmware Images anzeigen bzw. auswählen . . . 16
3.3.3 fwrm - Firmware Images löschen . . . 16
3.3.4 fwuncompress - Firmware Image auspacken . . . 17
3.4 Benutzerverwaltung . . . 18
3.5 Netzwerk Konfiguration . . . 19
3.5.1 netconfig - Konfigurationsänderungen für Netzwerkschnittstellen prüfen und anwenden . 19
3.5.2 nicinfo - Informationen über physikalische Netzwerkinterfaces abrufen . . . 20
3.5.3 nicmgr - Verwaltung von physischen Netzwerkschnittstellen . . . 22
3.5.4 netinfo - Informationen über logische/virtuelle Netzwerkinterfaces abrufen . . . 23
3.6 Systemdienste . . . 24
3.6.1 status - Systemdienst-Status anzeigen . . . 24
3.6.2 start - Systemdienst starten . . . 25
3.6.3 stop - Systemdienst stoppen . . . 25
3.6.4 restart - Systemdienst neu starten . . . 26
3.6.5 reload - Systemdienst neu laden . . . 26
3.6.6 svcconfig - Konfigurationsänderungen für Systemdienste prüfen und anwenden . . . 27
3.7 Systemstatus Anzeige - show und monitor . . . 28
3.7.1 cpuload - Anzeige der Prozessorauslastung . . . 28
3.7.2 devices - Auflistung von Meinberg Modulen und Baugruppen . . . 29
3.7.3 ip - Anzeige der IPv4 und IPv6 Adressen . . . 30
3.7.4 lantimelog - LANTIME Protokolldatei anzeigen . . . 31
3.7.5 linkstate - Anzeige des Verbindungsstatus von Netzwerkschnittstellen . . . 32
3.7.6 modules - Auflistung der geladenen Treiber bzw. Kernel Module . . . 33
3.7.7 network - Anzeige des aktuellen Netzwerkstatus . . . 34
3.7.8 processes - Liste der aktiven Prozesse . . . 35
3.7.9 route - Anzeige der IPv4 und IPv6 Routingtabelle . . . 38
3.7.10 syslog - System Protokolldatei anzeigen . . . 39
3.7.11 version - Anzeige der Firmware Version . . . 40
3.8 Systembefehle . . . 41
3.8.1 reboot - System neu starten . . . 41
3.8.2 make_noise - Gerät durch Blinken und Piepen identifizieren . . . 42

4 Texteditoren **43**
4.1 nano 43
4.2 edit 45
4.3 vim 46

1 Einleitung

Die Konfiguration und das Management eines Meinberg LANTIME Zeitservers kann über eine Reihe von Benutzer-Schnittstellen erfolgen. Die grafische Oberfläche des Geräts ist per Web-Browser erreichbar und bietet neben dem Anschauen und Verändern von Konfigurationseinstellungen vielfältige Möglichkeiten, sich den aktuellen Status des Gerätes bzw. statistische Daten mithilfe des grafischen Statistik-Tools xtrastats in Form von Diagrammen und Grafiken anzeigen zu lassen.

Das Command Line Interface (CLI) der sechsten LANTIME Firmware Generation (LTOS6) basiert hingegen auf einem reinen textbasierten Konzept. Es ist sowohl über lokale Schnittstellen (serielle Konsole) als auch aus der Ferne über Netzwerk (SSH, Telnet) erreichbar. Als Grundlage und Kommandozeilen-Interpreter wird die Bourne Again Shell (Bash) verwendet, die komfortable Eingaberoutinen bietet. Das Editieren von Befehlszeilen ist durch die Verwendung der Cursor-Tasten sowie der Entfernen- und Rückschritt-Taste („Backspace“) genau so möglich, wie das Aufrufen von bereits eingegebenen Befehlen, die dann verändert oder einfach erneut ausgeführt werden können. Mit der Tab-Taste können Befehle teilweise automatisch vervollständigt werden, ohne dass man sie komplett eingeben muss.

Die Verwendung einer Standard-Shell hat darüberhinaus noch weitere Vorteile. Ein Unix Systemadministrator ist die Verwendung einer solchen Shell bereits von anderen, Unix-basierten Systemen bekannt. Außerdem können die Skript-Sprachelemente der Shell genutzt werden, um wiederkehrende Aufgaben oder umfangreiche Befehlssequenzen zu automatisieren.

Neben der Standard-Bash ist auf dem System noch eine Posix kompatible Shell namens Dash (Debian Almquist Shell) sowie die Almquist Shell („ash“) installiert und kann genutzt werden.

Dieses Referenz-Handbuch beschreibt nicht alle über 400 Befehle, die auf einem LTOS6 System zur Verfügung stehen. Es beschränkt sich darauf, die CLI Befehle vorzustellen, die am häufigsten von LANTIME Administratoren verwendet werden. Viele Befehle sind aus dem Unix- bzw. GNU Linux Umfeld bekannt, bei einigen kann man über den Aufruf „befehlsname -h“ eine Hilfestellung angezeigt bekommen, wie der entsprechende Befehl verwendet werden kann.

Bei Fragen zur Verwendung des CLI auf Ihrem LANTIME wenden Sie sich bitte wie gewohnt an Ihren technischen Support.

2 Verwendung des CLI

Um das CLI zu verwenden, müssen Sie sich an einer der Benutzerschnittstellen anmelden, die den CLI-Zugriff ermöglichen. Das ist entweder ein serieller Terminal-Port des Gerätes oder TELNET bzw. SSH. Falls nur ein Webbrowser zur Verfügung steht, kann das CLI auch per Browser über den sogenannten WEBSHELL Dienst verwendet werden.

Serielle Konsole

Serielle Konsolenports sind entweder am Front Panel oder (bei modularen Systemen) am CPU Modul vorhanden, eventuell auch beides. Die seriellen Ports werden standardmässig mit einer Geschwindigkeit von 38400 Baud, 8 Datenbits, keine Parität, 1 Stopbit (8N1) angesprochen.

Anmeldung

Bei Auslieferung gibt es auf dem System nur einen Benutzer (root) mit einem Standardpasswort (timeserver). Andere Benutzer können das CLI nur verwenden, wenn sie die Zugriffsstufe „Super-User“ haben. Ist das nicht der Fall, wird ein Anmelden nicht erlaubt.

Bei Anmeldung eines „Super User“s wird diesem beim Anmelden ein Systemstatus angezeigt, danach wird der Befehlsprompt ausgegeben.

Automatische Abmeldung

Das CLI wird einen Benutzer automatisch abmelden, wenn dieser länger als 5 Minuten an der Eingabeaufforderung keinen Befehl eingibt. Dieses Timeout kann durch den Befehl „**no_shell_timeout**“ abgeschaltet werden und mit dem Befehl „**set TMOUT=x**“ auf x Sekunden geändert werden.

Eingabe von Befehlen

Bei der Eingabe von CLI Befehlen wird Gross/Kleinschreibung unterschieden. Es ist möglich, den Anfang eines Befehls einzugeben und ihn dann mit der TAB-Taste (STRG+I) vervollständigen zu lassen. Ist der eingegebene Teil nicht eindeutig, wird eine Liste aller passenden Befehle ausgegeben.

Um eine Befehlszeile zu bearbeiten, kann man mit den Cursor-Tasten links/rechts (alternativ auch STRG+B und STRG+F) bewegen. Die Eingabe von ESC+F bzw. ESC+B springt an den Anfang des nächsten bzw. vorherigen Wortes in einer Eingabezeile. Mit Backspace (STRG+H) löscht man das Zeichen links vom Cursor.

Befehlshistorie

Um bereits eingegebene Befehle erneut aufzurufen, können die Cursor-Tasten Hoch/Runter (STRG+P und STRG+N) verwendet werden. Man kann eine eingegebene Befehlszeile auch suchen, indem man STRG+R drückt und dann einen Suchbegriff eingibt. Durch erneutes Drücken von STRG+R kann dann in den passenden Befehlszeilen geblättert werden. Eine Liste aller eingebenene Befehle wird mit dem „**history**“ Befehl angezeigt.

Abmeldung, Beenden einer CLI Sitzung

Um sich vom CLI abzumelden, kann der Befehl „**exit**“ verwendet werden, wahlweise auch „**logout**“ oder STRG+D.

3 Befehlsreferenz

In diesem Kapitel werden die verschiedenen vom CLI zur Verfügung gestellten Befehle mit ihren benötigten Parametern erklärt.

Darstellungskonventionen

Die Befehlsnamen in diesem Kapitel werden **fett** dargestellt, etwaige Parameter sind *kursiv* hervorgehoben. Ist ein Parameter optional, d.h. man kann ihn weglassen, dann wird er in eckigen Klammern *[ ]* dargestellt.

Das Zeichen „#“ am Zeilenanfang stellt die Eingabeaufforderung (Prompt) dar, die sich je nach Konfiguration und System unterscheiden kann.

Beispiele:

pwd
(zeigt den Namen des aktuellen Verzeichnisses an)

ls [**path**]
(zeigt den Inhalt des angegebenen Verzeichnisses *[path]* an oder - wenn *[path]* weggelassen wurde, den Inhalt des aktuellen Verzeichnisses)

3.1 Konfigurationsverwaltung

Die LANTIME Firmware der 6. Generation bietet eine Reihe von Funktionen, um die Konfiguration des Gerätes zu verwalten. Dazu gehört das Speichern eines Konfigurationszustands („Config Set“) unter einem beliebigen Namen, das Laden/Aktivieren eines solchen gespeicherten Sets sowie Befehle, mit denen man prüfen kann, ob und wie eine gerade verwendete Konfiguration sich von der Start-Konfiguration des Gerätes unterscheidet.

3.1.1 lsconfig - Anzeige von gespeicherten Configsets

Zweck

Dieses Kommando zeigt alle für ein Package in einem Configset vorhandenen Dateien an oder die Configsets, in denen für ein Package eine Konfiguration vorhanden ist.

Aufruf und Parameter

lsconfig *[package] [configset]*

Der Parameter *package* gibt das Package an, für das Sie die verfügbaren Configsets bzw. die in einem Configset enthaltenen Dateien sehen möchten. Dabei werden die Dateien dann angezeigt, wenn sowohl das Package als auch *configset* angegeben wurden. Wird nur ein Package angegeben (und der Parameter configset wird weggelassen), dann zeigt lsconfig nur die Namen der Configsets an, in denen sich eine Konfiguration für das Package findet. Wird als Packagename „all“ angegeben, zeigt lsconfig die Dateien bzw. vorhandenen Configsets für alle Packages an.

Beispiele

```
# lsconfig network myconfig1
```

(zeigt die Namen aller für das Package „network“ gespeicherten Dateien im Konfigurationsstand „myconfig1“ an)

```
# lsconfig lantime
```

(zeigt alle Configsets an, in denen sich Konfigurationsdateien für das Package „lantime“ befinden)

```
# lsconfig all startup
```

(zeigt alle Dateinamen für alle Packages im Configset „startup“ an)

3.1.2 rmconfig - Löschen eines Configsets

Zweck

Dieser Befehl löscht eine vorher gespeicherte Konfiguration (d.h. ein „configset“). Achtung: rmconfig fragt nicht nach einer Bestätigung, sondern löscht sofort die angegebene Konfiguration dauerhaft aus dem Flash Speicher. Das ist nicht rückgängig zu machen, daher sollten Sie vorsichtig sein, wenn Sie diesen Befehl verwenden.

Aufruf und Parameter

rmconfig *package configset*

Der Parameter *package* gibt das Package an, dessen Konfiguration gelöscht werden soll. Das kann z.B. „network“ oder „lantime“ oder „snmp“ sein. Die vorhandenen Packages finden Sie, indem Sie sich den Inhalt des Verzeichnisses /package anschauen. Hier gibt es für jedes verfügbare Package ein entsprechend benanntes Unterverzeichnis. Wird als Package „all“ angegeben, wird der gesamte Konfigurationsstand gelöscht.

Mit dem Parameter *configset* wird spezifiziert, aus welchem Konfigurationssatz („configset“) die Konfiguration für das Package gelöscht werden soll. Dabei ist es nicht erlaubt, „default“ als configset anzugeben, weil die Default-Konfiguration eines Packages nicht gelöscht werden darf. Gibt man hier als configset „startup“ an, wird die Konfiguration für das Package somit wieder in den Auslieferungszustand versetzt bzw. beim nächsten Neustart wird diese Default-Konfiguration wieder verwendet.

Beispiele

```
# rmconfig network myconfig1
```
(löscht die Netzwerkkonfiguration im gespeicherten Konfigurationsstand „myconfig1“)

```
# rmconfig snmp startup
```
(löscht die Startup-Konfiguration des SNMP Packages, somit wird beim nächsten Neustart die Default-Konfiguration geladen)

```
# rmconfig all myconfig2
```
(löscht den gesamte Konfigurationssatz „myconfig2“, d.h. die darin gespeicherte Konfiguration für alle Packages)

3.1.3 diffconfig - Ungespeicherte Änderungen anzeigen

Zweck

Mit diesem Befehl kann man sich anzeigen lassen, welche noch nicht gespeicherten Konfigurationsänderungen vorliegen.

Aufruf und Parameter

diffconfig *package configset*

Der Parameter *package* gibt das Package an, dessen Konfiguration verglichen werden soll, z.B. „snmp“ oder „network“. Die vorhandenen Packages finden Sie, indem Sie sich den Inhalt des Verzeichnisses /packages anschauen. Hier gibt es für jedes verfügbare Package ein entsprechend benanntes Unterverzeichnis. Wird als Package „all“ angegeben, wird der gesamte Konfigurationsstand zum Vergleich herangezogen.

Mit dem Parameter *configset* kann man angeben, mit welchem Konfigurationssatz („configset“) die aktuelle Konfiguration abgeglichen werden soll. Wird kein Configset angegeben, wird „startup“ angenommen.

Beispiele

```
# diffconfig network myconfig1
```
(vergleicht die aktuelle Netzwerkkonfiguration mit dem gespeicherten Konfigurationsstand „myconfig1“)

```
# diffconfig all myconfig2
```
(vergleicht alle Konfigurationsdateien mit dem Configset „myconfig2“, d.h. die darin gespeicherte Konfiguration für alle Packages)

3.1.4 checkconfig - Prüfung auf ungespeicherte Konfigurationsänderungen

Zweck

Mit diesem Befehl kann man sich anzeigen lassen, ob noch nicht gespeicherte Konfigurationsänderungen vorliegen. Im Gegensatz zu **diffconfig** wird **checkconfig** schneller ausgeführt, zeigt dafür aber nicht die Änderungen im Detail an.

Aufruf und Parameter

checkconfig

Dieser Befehl muss ohne Parameter gestartet werden.

Beispiele

```
# checkconfig
```
No configuration changes.

3.1.5 saveconfig - Konfigurationsänderungen speichern

Zweck

Dieser Befehl speichert den aktuellen Zustand der Konfiguration in einem „configset“. Damit können Konfigurationsänderungen persistent gemacht werden (d.h. sie werden bei einem Neustart verwendet) und es kann eine Sicherung der momentan laufenden Konfiguration durchgeführt werden.

Aufruf und Parameter

saveconfig *package configset*

Mit *package* gibt man das Package an, dessen Konfiguration gespeichert werden soll. Hier kann z.B. „network“ oder „lantime“ oder „snmp“ angegeben werden. Wird als Package „all“ angegeben, wird die gesamte Konfiguration gespeichert (das ist auch das Standardverhalten, wenn man keine Parameter angibt).

Der Parameter *configset* gibt an, in welches Konfigurationssatz („configset“) die Konfiguration für das Package gespeichert werden soll. Dabei ist es nicht erlaubt, „default“ als configset anzugeben, weil die Default-Konfiguration eines Packages nicht überschrieben werden kann. Wird kein configset angegeben, wird „startup“ angenommen, und somit wird die gespeicherte Konfiguration beim nächsten Neustart verwendet.

Ist in dem angegebenen Konfigurationssatz bereits eine Konfiguration für das gewählte Package enthalten, wird diese aktualisiert.

Beispiele

```
# saveconfig snmp
```
(speichert die SNMP Konfiguration im „startup“ Konfigurationsstand)

```
# saveconfig network backup1
```
(speichert die aktuelle Netzwerk Konfiguration im Configset „backup1“)

```
# saveconfig all myconfig2
```
(speichert die gesamte Konfiguration in „myconfig2“, d.h. für alle Packages)

3.1.6 loadconfig - Configset laden

Zweck

Dieser Befehl lädt eine in einem „configset“ gespeicherte Konfiguration. Damit können gesicherte Konfigurationsstände wieder hergestellt werden (d.h. sie werden in das laufende System geladen).

Aufruf und Parameter

loadconfig *package configset*

Mit *package* gibt man das Package an, dessen Konfiguration geladen werden soll. Hier kann z.B. „network“ oder „lantime“ oder „snmp“ verwendet werden. Wird als Package „all“ angegeben, wird die Konfiguration aller Packages geladen (das ist auch das Standardverhalten, wenn man keine Parameter angibt).

Der Parameter *configset* gibt an, aus welchem Konfigurationssatz („configset“) die Konfiguration für das Package geladen werden soll. Wird hier „default“ angegeben, wird die Default-Konfiguration geladen. Soll die geladene Konfiguration persistent verwendet werden, muss sie nach dem loadconfig mit „saveconfig“ als startup-Konfiguration gespeichert werden.

Beispiele

```
# loadconfig snmp
```
(lädt die „startup“ SNMP Konfiguration, d.h. den Konfigurationsstand der bei einem Neustart verwendet würde)

```
# loadconfig network backup1
```
(Stellt die aktuelle Netzwerk Konfiguration aus dem Configset „backup1“ wieder her)

```
# loadconfig all myconfig2
```
(stellt die gesamte Konfiguration aus „myconfig2“ wieder her)

3.2 Datei Verwaltung

Die Verwaltung der Dateien im Flash Speicher eines LANTIME Gerätes wird normalerweise von der Firmware selbst erledigt. In Ausnahmesituationen kann es aber nötig sein, dass ein Administrator zum Beispiel eine Datei löschen, kopieren oder umbenennen muss. Es kann auch sein, dass Sie von Zeit zu Zeit den Inhalt einer Datei anschauen müssen (vielleicht eine Log- oder Status-Datei).

Im Folgenden werden CLI-Befehle beschrieben, mit denen Sie diese Aufgaben erledigen können.

3.2.1 pwd - Arbeitsverzeichnis anzeigen

Zweck

Mit diesem Befehl kann man abfragen, in welchem Verzeichnis man sich gerade befindet („print working directory“).

Aufruf und Parameter

pwd

Es gibt keine Parameter.

Beispiele

```
 # pwd snmp
/var/run
```

3.2.2 cd - Arbeitsverzeichnis wechseln

Zweck

Mit diesem Befehl kann man in ein angegebenes Verzeichnis wechseln („change directory“).

Aufruf und Parameter

cd *[Verzeichnisname]*

Das System wechselt in das angegebene Verzeichnis oder, wenn kein Verzeichnis angegeben wurde, in das Home-Verzeichnis des Benutzers.

Beispiele

```
# cd /etc
```
(wechselt in das /etc Verzeichnis)

```
# cd
```
(wechselt in das Home Verzeichnis des Benutzers, z.B. /root für den Root Benutzer)

3.2.3 ls - Verzeichnisinhalt anzeigen

Zweck

Der „ls“ Befehl (list) zeigt alle in einem Verzeichnis enthaltenen Unterverzeichnisse und Dateien an.

Aufruf und Parameter

ls *[Optionen] [Verzeichnisname]*

Der Inhalt des Verzeichnisses [**Verzeichnisname**] wird aufgelistet. Es stehen eine Vielzahl von Optionen zur Verfügung, mit denen das Ausgabeformat ausgewählt werden kann. Mit der Option „–help“ kann eine Auflistung aller möglichen Optionen angezeigt werden.

Wird kein Verzeichnisname angegeben, wird der Inhalt des aktuellen Arbeitsverzeichnisses ausgegeben.

Beispiele

```
# ls /var/log
```
(zeigt den Inhalt des Verzeichnisses /var/log an im Standardausgabeformat an)

```
# ls -l /var/run
```
(zeigt den Inhalt des Verzeichnisses /var/run im Lang-Format (-l) an, das einige Details wie z.B. Grösse der Datei mit ausgibt)

```
# ls
```
(zeigt den Inhalt des aktuellen Arbeitsverzeichnisses an)

3.2.4 cp - Dateien und/oder Verzeichnisse kopieren

Zweck

Der „cp“ Befehl kann Dateien oder ganze Verzeichnisse kopieren.

Aufruf und Parameter

cp *[Optionen] [Quelle(n [Ziel]*

Eine Liste aller möglichen Optionen kann mit

```
cp -help
```

aufgerufen werden. Die Option „-v“ („verbose“) zeigt beim Kopieren an, welche Datei gerade bearbeitet wird.

Als Quelle(n) können eine oder mehrere Dateien angegeben werden, hier sind auch Platzhalter (Wildcards) möglich. Als Ziel kann ein Zielverzeichnis angegeben werden, in das die Quelldateien kopiert werden. Wird nur eine Datei als Quelle angegeben, ist als Ziel auch die Angabe eines Zielverzeichnisses plus Zieldateinamen möglich.

Soll ein ganzes Verzeichnis inklusive aller Unterverzeichnisse und enthaltenen Dateien kopiert werden, muss die „-r“ Option verwendet werden.

Beispiele

```
# cp /etc/hosts /var/tmp
```

(kopiert die Datei hosts aus dem /etc Verzeichnis in das Zielverzeichnis /var/tmp, dort wird sie unter gleichem Namen gespeichert, also hosts)

```
# cp /config/global_configuration /var/tmp/mycopy
```

(kopiert die Datei global_configuration aus dem Verzeichnis /config in das Zielverzeichnis /var/tmp unter dem Dateinamen mycopy)

```
# cp /etc/ssh/ssh_* /tmp/
```

(kopiert alle Dateien aus /etc/ssh, deren Dateiname mit „ssh_“ anfängt, in das Verzeichnis /tmp)

```
# cp -r /etc/udev /tmp/
```

(kopiert das Verzeichnis /etc/udev inklusive aller Unterverzeichnisse und enthaltenen Dateien in das Zielverzeichnis /tmp)

3.2.5 mv - Dateien und/oder Verzeichnisse verschieben bzw. umbenennen

Zweck

Mit dem „mv“ Befehl kann man Dateien oder ganze Verzeichnisse an eine andere Stelle verschieben bzw. umbenennen.

Aufruf und Parameter

mv *[Optionen] [Quelle(n [Ziel]*

Eine Liste aller möglichen Optionen kann mit

```
mv -help
```

aufgerufen werden.

Als Quelle(n) können eine oder mehrere Dateien angegeben werden, hier sind auch Platzhalter (Wildcards) möglich. Als Ziel kann ein Zielverzeichnis angegeben werden, in das die Quelldateien verschoben werden. Wird nur eine Datei als Quelle angegeben, kann als Ziel auch die Angabe eines Zielverzeichnisses plus Zieldateinamen dazu verwendet werden, um eine Datei umzubenennen.

Soll ein ganzes Verzeichnis inklusive aller Unterverzeichnisse und enthaltenen Dateien kopiert werden, kann das Verzeichnis als Quelle angegeben werden.

Beispiele

```
# mv /dir_a/file_a /dir_b/
```

(verschiebt die Datei file_a aus dem /dir_a Verzeichnis in das Zielverzeichnis /dir_b)

```
# mv /dir_a/file_a /dir_b/file_b
```

(verschiebt die Datei file_a aus dem Verzeichnis /dir_a in das Zielverzeichnis /dir_b und benennt sie um in file_b)

```
# mv /dir_a/file_*.txt /tmp/
```

(verschiebt alle Dateien aus /dir_a, deren Dateiname mit „file_“ anfängt und mit „.txt“ aufhört, in das Verzeichnis /tmp)

```
# mv /dir_a /tmp/
```

(verschiebt das Verzeichnis /dir_a inklusive aller Unterverzeichnisse und enthaltenen Dateien in das Zielverzeichnis /tmp)

3.2.6 rm - Dateien und/oder Verzeichnisse löschen

Zweck

Der „rm“ Befehl erlaubt das Löschen von Dateien oder ganzen Verzeichnissen. Das kann dazu führen, dass ein System nicht mehr korrekt funktioniert, Sie sollten diesen Befehl daher nur einsetzen, wenn Sie sich sicher sind, dass die zu löschende Datei bzw. das zu löschende Verzeichnis nicht für den ordnungsgemässen Betrieb Ihres LANTIME Systems benötigt wird. Im Zweifel fragen Sie bitte den Meinberg Support.

Die gelöschten Dateien und Verzeichnisse können nicht wiederhergestellt werden und sind unwiederbringlich verloren, es ist daher äusserste Vorsicht geboten.

Aufruf und Parameter

rm *[Optionen] [Datei1] [Datei2] ...*

Eine Liste aller möglichen Optionen kann mit

```
rm –help
```

aufgerufen werden.

Es können eine oder mehrere Dateien angegeben werden, hier sind auch Platzhalter (Wildcards) möglich. Um ein gesamtes Verzeichnis inklusive aller Unterverzeichnisse zu löschen, kann die Option „-r“ angegeben werden.

Im Normalfall erfolgt keine Sicherheitsabfrage, bitte wenden Sie diesen Befehl also nur auf Dateien und Verzeichnisse an, von denen Sie sicher sind dass sie nicht mehr benötigt werden.

Beispiele

```
# rm /dir_a/file_a
```

(löscht die Datei file_a aus dem /dir_a Verzeichnis)

```
# rm -r  /dir_b/
```

(löscht das Verzeichnis /dir_b sowie alles was sich darin befindet - inklusive aller Unterverzeichnisse und deren Inhalt - für immer)

3.2.7 cat - Datei anzeigen

Zweck

Mit „cat“ kann man den Inhalt einer Datei auf dem Bildschirm anzeigen lassen.

Aufruf und Parameter

cat *[Dateiname]*

Der Inhalt der Datei [**Dateiname**] wird ausgegeben. Es stehen eine Vielzahl von Optionen zur Verfügung, mit denen das Ausgabeformat ausgewählt werden kann.

Wird keine Datei angegeben, wird die Standardeingabe verwendet. Es ist möglich, die Ausgabe von „cat“ mit dem Befehl „less“ zu kombinieren, um eine Datei seitenweise anzuzeigen.

Beispiele

```
# cat /var/log/messages
```
(zeigt den Inhalt der Datei messages im Verzeichnis /var/log an)

```
# cat /var/log/lantime_messages | less
```
(zeigt den Inhalt der Datei lantime_messages im Verzeichnis /var/log an, der „less“ Befehl sorgt dafür, dass man innerhalb der Datei blättern kann (hoch/runter, Leertaste=nächste Seite, „/“=Suche, Verlassen mit „q“).

3.3 Firmware Verwaltung

LTOS V6 bietet die Möglichkeit, mehrere Firmwarestände parallel auf dem LANTIME zu installieren. Es kann dann ausgewählt werden, welcher dieser sogenannten Firmware-Images beim nächsten Systemstart verwendet werden soll. Desweiteren kann ein Firmware Image auch anstatt über das Web Interface geöscht bzw. manuell per CLI installiert werden. Die dafür benötigten Befehle werden nachfolgend beschrieben.

3.3.1 fwlist - Installierte Firmware Images anzeigen

Zweck

Mit dem „fwlist“ Befehl kann eine Liste aller installierten Firmware Images angezeigt werden.

Aufruf und Parameter

fwlist *[-v] [Suchpattern]*

Mit *Suchpattern* kann ein Dateinamenmuster angegeben werden, das zur Filterung der auszugebenden Liste verwendet werden kann. Wird kein Suchpattern angegeben, werden alle installierten Images angezeigt.

Wird die Option „-v“ angegeben, werden hinter den FW-Image Namen die Versionsnummern sowie der Image-Typ angezeigt. Ein Firmware Image kann entweder ein „Standard“ Image sein oder ein „Compressed“ Image. Letzteres benötigt weniger Speicher und es können keine Dateien im Image geändert oder gelöscht werden und ein Hinzufügen von Dateien ist ebenfalls nicht möglich. Um ein „Compressed“ Image

Beispiele

```
# fwlist
```
(zeigt alle installierten Firmware-Images an)

```
# fwlist fw_*
```
(zeigt nur die installierten Firmware-Images an, deren Name mit „fw_“ beginnt)

```
# fwlist OSV
```
(zeigt nur das installierten Firmware-Image an, dessen Name „OSV“ ist)

3.3.2 fwselect - Aktives Firmware Images anzeigen bzw. auswählen

Zweck

Der „fwselect“ Befehl erlaubt das Auswählen des aktiven Firmware Images (wird nach einem Reboot aktiv) oder zeigt an, welches Firmware Image gerade ausgewählt ist.

Aufruf und Parameter

fwselect *[FWImage]*

Wird der *FWImage* Parameter angegeben, versucht das System, ein entsprechend benanntes installiertes Firmware Image zu finden und zu aktivieren, d.h. es wird für den nächsten Systemstart vorgesehen. Kommt es beim Aktivieren eines Images zu Fehlern, wird der Ursprungszustand wieder hergestellt.

Ruft man „fwselect“ ohne einen Parameter auf, zeigt es an, welches Firmware Image aktiviert ist, d.h. beim nächsten Systemstart gebootet wird.

Beispiele

```
# fwselect
```
(zeigt das aktivierte Firmware-Image an)

```
# fwselect fw_6.12.004
```
(aktiviert das Firmware Image mit dem Namen fw_6.12.004)

3.3.3 fwrm - Firmware Images löschen

Zweck

Mit dem „fwrm“ Befehl können nicht mehr benötigte Firmware Images gelöscht werden, solange sie nicht gerade aktiv sind oder für den nächsten Systemstart ausgewählt wurden (siehe fwselect).

Aufruf und Parameter

fwrm *[FWImage]*

oder

```
# fwrm [-wipe-all [keep=X]] [FWImage]
```

Der *FWImage* Parameter gibt an, welches Image gelöscht werden soll. Die zweite Aufrufform (–wipe-all) löscht alle Firmware Images außer dem OSV Image, die nicht gerade aktiv sind und auch nicht für den nächsten Systemstart mittels fwselect ausgewählt wurden. Durch den optionalen „keep“ Parameter kann die Anzahl der Images angegeben werden, die man zusätzlich zu den nicht löschbaren Images behalten möchte.

Die Option –wipe-all kann auch abgekürzt als -W geschrieben werden.

Beispiele

```
# fwrm fw_6.14.021
```
(löscht das Firmware-Image fw_6.14.021)

```
# fwrm -wipe-all keep=2
```
(löscht alle Firmware-Images außer dem gerade aktiven Image, dem OSV Image und einem für den nächsten Systemstart per fwselect ausgewählten Image)

3.3.4 fwuncompress - Firmware Image auspacken

Zweck

Ab Version 6.15 werden Firmware Images grundsätzlich in komprierter Form („compressed“) installiert, um Speicherplatz zu sparen. Ein solches Image ist read-only und kann nicht verändert werden. Das ist im normalen Betrieb auch nicht notwendig, kann aber im Einzelfall bei Anpassungen notwendig werden. Mit dem „fwuncompress“ Befehl kann man ein komprimiertes Firmware Image auspacken und ein nicht komprimiertes Image daraus erzeugen. Der „fwuncompress“ Befehl behält dabei das komprimierte Image und erzeugt ein neues Image. Wenn man das Ausgangsimage nicht mehr benötigt, kann man es anschließend mit „fwrm“ löschen (solange es sich dabei nicht um das aktuell laufende oder für den nächsten Systemstart ausgewählte Image handelt).

Das gerade aktive Image kann per „fwuncompress“ im laufenden Betrieb ausgepackt werden.

Aufruf und Parameter

fwuncompress *FWImage*

Das angegebene komprimierte Image (Name fängt meist mit „fw_“ an) wird verwendet, um eine unkomprimierte Kopie zu erzeugen, die dann ein „u“ vorangestellt bekommt. Aus „fw_6.15.015“ wird also „ufw_6.15.015“.

Beispiele

```
# fwuncompress fw_6.16.002
```

(packt das komprimierte Firmware Image fw_6.616.002 in ein neu erzeugtes Image namens ufw_6.16.002 aus)

3.4 Benutzerverwaltung

Das System unterstützt verschiedene lokale Benutzerkonten und eine Authentifizierung mittels eines externen Servers, z.B. mithilfe von RADIUS oder TACACS+. Für die Verwaltung der Benutzerkonten sowie für die Anzeige des aktuellen Status stehen eine Reihe von Befehlen zur Verfügung, die in diesem Kapitel vorgestellt werden.

3.5 Netzwerk Konfiguration

Neben der Möglichkeit, Einstellungen für die Netzwerkschnittstelle(n) eines LANTIMEs über das (eventuell vorhandene) Front-Panel oder - sobald die initiale Konfiguration vorgenommen wurde - über das Web Interface zu verändern, kann auch das CLI für die Veränderung der Netzwerkparameter eingesetzt werden. Das kann zum Beispiel wichtig sein, wenn das Gerät per Netzwerk nicht mehr erreichbar ist oder wenn Einstellungen vorgenommen werden müssen, die im Web Interface nicht vorgesehen sind.

Hauptsächlich wird die Netzwerkkonfiguration in der Konfigurationsdatei /etc/mbg/net.cfg festgehalten, die mithilfe eines Editors (siehe entsprechendes Kapitel) bearbeitet werden kann. Für die Anwendung der Einstellungen wird dann der Befehl „**netconfig**“ aufgerufen.

3.5.1 netconfig - Konfigurationsänderungen für Netzwerkschnittstellen prüfen und anwenden

Zweck

Der **netconfig** Befehl vergleicht die Netzwerk-relevante Konfiguration mit dem aktuellen Status und führt eventuell notwendige Schritte aus, um den konfigurierten Zustand zu erreichen. Ist z.B. ein virtuelles Netzwerkinterface zwar konfiguriert aber noch nicht vorhanden, wird es von netconfig automatisch angelegt und initialisiert (z.B. mit einer statischen IP Adresse or durch den Start eines DHCP Clients).

Als Basis für die Netzwerk-Konfiguration wird die Konfigurationsdatei */etc/mbg/net.cfg* verwendet, deren Aufbau und Inhalt im Kapitel Konfigurationsdateien genauer beschrieben wird.

Aufruf und Parameter

netconfig

Der Befehl hat keine Parameter.

Beispiele

```
# netconfig
```

(überprüft alle Netzwerkschnittstellen auf notwendige Konfigurationsänderungen und wendet diese an)

3.5.2 nicinfo - Informationen über physikalische Netzwerkinterfaces abrufen

Zweck

Mit **nicinfo** kann man sich den aktuellen Status von physikalischen Netzwerkinterfaces anzeigen lassen, d.h welche MAC Adresse, , welche Bonding-Gruppe, welcher Linkmode (z.B. 100MBit/s Full Duplex) und welche IPV6 Einstellungen eine Netzwerkschnittstelle hat.

Aufruf und Parameter

nicinfo *[Optionen] [INTERFACE]*

Der Befehl versteht folgende Optionen:
-c Check Link Mode
Zeigt nur den aktuelle Linkstatus (100FDX) an sowie die Einstellung ob die Schnittstelle überwacht wird (LINK_CHECK).

-s Short Mode
Diese Option führt dazu, dass keine Detailinformationen sondern nur der momentane Konfigurationsstatus der Schnittstelle angezeigt wird:

```
+ Nicht vorhanden, muss angelegt werden
! Geändert, muss konfiguriert werden
- Vorhanden aber nicht konfiguriert, muss entfernt werden
= Entspricht der Konfiguration
```

Wird als INTERFACE Parameter eine Interface-Bezeichnung (z.B. lan0) angegeben, wird nur dieses Interface angezeigt. Lässt man INTERFACE weg bzw. leer, werden alle Interfaces angezeigt.

Beispiele

```
# nicinfo

Please wait ...
Current state of physical interfaces:
lan0 matches configuration (lan0 00:13:95:00:6b:ef - 100FDX AUTO=ON
IPV6=ACTIVATED+AUTOCONF)
lan1 matches configuration (lan1 00:60:6e:7a:d3:4d - 10HDX AUTO=ON
IPV6=ACTIVATED)
lan2 matches configuration (lan2 00:60:6e:7a:d3:4e - 10HDX AUTO=ON
IPV6=DEACTIVATED)
lan3 matches configuration (lan3 00:60:6e:7a:d3:4f - 10HDX AUTO=ON
IPV6=DEACTIVATED)
```

(zeigt den Status aller physikalischen Netzwerkschnittstellen an)

```
# nicinfo -s
=lan0
!lan1/1
=lan2
=lan3
```

(zeigt den Konfigurationsstatus aller physikalischen Netzwerkschnittstellen an, im Beispiel liegen Änderungen für lan1 vor)

```
# nicinfo -c lan0
Please wait ...
Current state of lan0:
Status of physical interface lan0 is 100FDX LINK_CHECK=ON
```

(zeigt den Link Status von lan0 an sowie die Konfigurationseinstellung für die Netzwerküberwachung, in diesem

Fall ist sie für lan0 eingeschaltet)

3.5.3 nicmgr - Verwaltung von physischen Netzwerkschnittstellen

Zweck

Mit dem „nicmgr“ Befehl kann man neu hinzugefügte Netzwerkschnittstellen (z.B. durch ein neues LNE Modul) am System anmelden oder ausgefallene/ausgetauschte Interfaces entfernen. Vor dem Anmelden von neuen Netzwerkports sind diese nicht nutzbar, d.h. sie können nicht konfiguriert werden.

Aufruf und Parameter

nicmgr help

oder

nicmgr assign [FREE_IF] [IFNUMBER]

oder

nicmgr remove [IFNUMBER]

oder

nicmgr autoassign

oder

nicmgr autoreplace

Dabei repräsentiert FREE_IF ein noch nicht angemeldetes Interface. Diese Interfaces haben die Bezeichnung ethX, wobei X eine fortlaufende Nummer ist. Bei einem neu eingesteckten LNE Modul mit 4 Netzwerkports werden diese als eth0, eth1, eth2 und eth3 im System erscheinen und man kann sie dann im System anmelden, indem man sie einem neuen physischen Port zuweist. Angemeldete Interfaces werden unter der Bezeichnung lanX geführt, wobei X eine eindeutige und im Regelfall fortlaufende Nummer ist. Der erste Netzwerkport im CPU Modul wird z.B. immer als lan0 bezeichnet.

IFNUMBER bezeichnet die Nummer eines anmeldeten Ports, also entspricht die „1“ dem physischen Netzwerk Interface lan1, die „5“ steht für lan5 und so weiter.

Die beiden Kommandos „autoassign“ und „autoreplace“ erleichtern die Zuweisung/den Austausch von mehreren Ports. Während „autoassign“ alle nicht zugewiesenen eth%X Ports als neue Ports anmeldet, sucht „autoreplace“ nach angemeldeten Interfaces die nicht mehr verfügbar sind (also z.B. ausgefallen sind oder entfernt wurden) und ersetzt diese mit nicht zugewiesenen ethX Ports, sofern vorhanden.

Beispiele

```
# nicmgr assign eth0 7
```
(meldet den nicht zugewiesenen Netzwerkport eth0 als lan7 im System an)

```
# nicmgr remove 6
```
(meldet lan6 im System ab)

```
# nicmgr autoassign
```
(meldet alle noch nicht zugewiesenen Netzwerkports automatisch im System an)

```
# nicmgr autoreplace
```
(nicht zugewiesene ethX Ports werden nicht mehr vorhandenen angemeldeten Netzwerkports automatisch zugewiesen)

3.5.4 netinfo - Informationen über logische/virtuelle Netzwerkinterfaces abrufen

Zweck

Der Befehl **netinfo** ermöglicht es, sich den aktuellen Status von logischen Netzwerkinterfaces anzeigen lassen, d.h IP Adressen und Konfigurationsstatus (also ob z.B. der aktuelle Zustand dem konfigurierten entspricht).

Aufruf und Parameter

netinfo *[Optionen] [INTERFACE]*

Der Befehl versteht folgende Optionen:
-a Advanced Info Mode
Zeigt ausführlichere Informationen an, also z.B. die zugehörige MAC Adresse und den administrativen Status.

-s Short Mode
Diese Option führt dazu, dass keine Detailinformationen sondern nur der momentane Konfigurationsstatus angezeigt wird:

```
+ Nicht vorhanden, muss angelegt werden
! Geändert, muss konfiguriert werden
- Vorhanden aber nicht konfiguriert, muss entfernt werden
= Entspricht der Konfiguration
_ Keine Konfiguration vorhanden
```

-i ID Mode
Diese Option kann man mit -s kombinieren, dann wird neben dem Konfigurationsstatus und dem Interface-Namen auch noch die Interface Nummer angegeben. Ohne -s hat die Verwendung von -i keine Auswirkungen, d.h. die Ausgabe von netinfo ohne irgendwelche Optionen sowie die Ausgabe von netinfo -a sind identisch, egal ob -i angegeben wird oder nicht.

Wird als INTERFACE Parameter eine Interface-Bezeichnung (z.B. lan0:0) angegeben, wird nur dieses Interface angezeigt. Lässt man INTERFACE weg bzw. leer, werden alle Interfaces angezeigt.

Beispiele

```
# netinfo
Current state of logical interfaces:
bond0:2 matches configuration (STATIC 10.99.109.11 255.255.255.0 - NONE)
lan0:0 matches configuration (DHCP - - - NONE)
lan1:1 [Virtual Interface 1] is not active and requires to be configured
bond0:3 has no configuration
```

(zeigt den Status aller logischen Netzwerkschnittstellen an)

```
# netinfo -s
=bond0:2
=lan0:0
+lan1:1
_bond0:3
```

(wie oben, zeigt den Konfigurationsstatus aller logischen Netzwerkschnittstellen an)

```
# netcinfo -s -i lan0:0
=lan0:0/0
```

(zeigt den Konfigurationsstatus von lan0:0 an sowie die Nummer des logischen Interfaces (/0).

3.6 Systemdienste

Die auf dem System installierten Dienste erfüllen unterschiedliche Aufgaben. Der Grossteil dieser Services ist für die Bereitstellung von Netzwerkdiensten zuständig, zum Beispiel der SSH Dienst oder der HTTP Dienst. Die Kontrolle der Dienste, das heisst welcher Dienst wann gestartet oder gestoppt wird, wird im Normalfall automatisch anhand der Konfiguration des Systems gesteuert. Ist zum Beispiel TELNET auf allen Netzwerkinterfaces deaktiviert, wird der entsprechende Dienst automatisch gestoppt. Sobald TELNET auf einem Interface aber wieder vom Benutzer aktiviert wird, startet der Dienst automatisch.

In besonderen Situationen kann es allerdings notwendig sein, den Status eines Dienstes zu prüfen oder einen Dienst manuell zu starten oder zu stoppen. Nach manuellen Konfigurationsänderungen ist häufig ein Neustart eines Dienstes nötig, damit dieser die geänderte Konfiguration anwendet.

Mit dem Befehl „status all“ kann man sich den Status aller registrierten Systemdienste anschauen (siehe status Befehl) und damit auch gleichzeitig eine Liste aller verfügbaren Systemdienste zu erhalten.

Die im folgenden beschriebenen Befehle dienen der Kontrolle der Systemdienste.

3.6.1 status - Systemdienst-Status anzeigen

Zweck

Mit dem Kommando status können Sie sich anzeigen lassen, ob ein bestimmter Systemdienst gerade ausgeführt wird oder nicht. Auch das Anzeigen des Status aller Dienste ist möglich.

Aufruf und Parameter

status *[service]*

Mit *service* geben Sie den Dienst an, dessen Status angezeigt werden soll. Wenn Sie sich den Status aller Dienste anschauen möchten, geben Sie als Service-Namen „all“ an.

Beispiele

```
# status ssh
```
(zeigt an, ob der SSH Dienst ausgeführt wird oder nicht)

```
# status all
```
(zeigt den Status aller Systemdienste an)

3.6.2 start - Systemdienst starten

Zweck

Dieses Kommando startet einen Systemdienst, wenn er noch nicht läuft. Das reine Starten eines Dienstes führt nicht dazu, dass er von aussen erreichbar ist, dazu muss er für das jeweilige virtuelle Netzwerkinterface freigegeben werden. Das System startet einen Dienst automatisch, wenn er auf einem Interface freigegeben wurde und stoppt ihn wieder, sobald er auf keinem Interface mehr freigegeben ist.

Aufruf und Parameter

start *[service]*

Der Parameter *service* gibt an, welcher Dienst gestartet werden soll. Sollte der angegebene Systemdienst bereits laufen, passiert nichts. Den aktuellen Status eines Dienstes kann man mit dem **status** Befehl überprüfen.

Beispiele

```
# start ssh
```
(startet den SSH Dienst)

```
# start http
```
(startet den HTTP Dienst)

3.6.3 stop - Systemdienst stoppen

Zweck

Dieses Kommando stoppt einen Systemdienst, wenn er läuft. Nach dem Stoppen eines Dienstes ist er auf keinem Interface mehr aktiv, bis er mit **start** neu gestartet wird. Das System stoppt einen Dienst automatisch, wenn er auf keinem Interface mehr aktiv ist (gilt für netzwerkrelevante Dienste).

Aufruf und Parameter

stop *[service]*

Mit *service* geben Sie an, welcher Dienst gestoppt werden soll. Wird der angegebene Systemdienst schon nicht ausgeführt, passiert nichts. Den aktuellen Status eines Dienstes kann man mit dem **status** Befehl überprüfen.

Beispiele

```
# stop ssh
```
(stoppt den SSH Dienst - Vorsicht: Damit werden auch alle laufenden SSH Verbindungen beendet, sogar diejenige, die Sie für die Eingabe dieses Befehls verwendet haben)

```
# stop https
```
(stoppt den HTTPS Dienst)

3.6.4 restart - Systemdienst neu starten

Zweck

Dieses Kommando startet einen Systemdienst neu, d.h. wenn er läuft, wird er gestoppt und wieder gestartet. Läuft er nicht, wird er (ohne den Stop Befehl auszuführen) einfach gestartet.

Aufruf und Parameter

restart *[service]*

Mit dem Parameter *service* geben Sie an, welcher Dienst neu gestartet werden soll.

Beispiele

```
# restart ssh
```
(startet den SSH Dienst neu, etwaige laufende SSH Verbindungen werden unterbrochen)

```
# restart http
```
(startet den HTTP Dienst neu)

3.6.5 reload - Systemdienst neu laden

Zweck

Dieses Kommando veranlasst einen Systemdienst dazu, seine Konfiguration neu einzulesen und anzuwenden. Bei den meisten Systemdiensten ist das nur durch einen Restart (siehe Kommando **restart**) möglich, der „**reload**“ Befehl wendet dabei bei jedem Dienst jeweils die Methode an, die für diesen Dienst notwendig ist.

Aufruf und Parameter

reload *[service]*

Mit dem Parameter *service* geben Sie an, welcher Dienst neu gestartet werden soll.

Beispiele

```
# reload ssh
```
(Lädt die Konfiguration des SSH Dienstes neu (durch Neustart), etwaige laufende SSH Verbindungen werden unterbrochen)

```
# restart http
```
(lädt die Konfiguration des HTTP Dienstes neu (durch Neustart))

3.6.6 svcconfig - Konfigurationsänderungen für Systemdienste prüfen und anwenden

Zweck

Mit **svcconfig** wird das System angewiesen, für alle Systemdienste zu prüfen, ob es Konfigurationsänderungen gibt. Das wird dadurch erreicht, dass für einen Dienst registrierte Dateien auf Änderungen (seit dem letzten Start des Dienstes) überprüft werden. Die Liste der registrierten Dateien ist in /var/run/svccfg.db abgelegt und kann mit dem **cat** Befehl angeschaut werden.

Sollte eine der Dateien verändert worden sein, wird **svcconfig** den entsprechenden Dienst per **reload** Befehl dazu bringen, seine Konfiguration neu einzulesen.

Aufruf und Parameter

svcconfig

Der Befehl hat keine Parameter. Es werden immer alle registrierten Dateien für alle Dienste überprüft.

Beispiele

```
# svcconfig
```

(überprüft alle Dienste auf Konfigurationsänderungen und startet/lädt einen Dienst neu, wenn Änderungen erkannt werden)

3.7 Systemstatus Anzeige - show und monitor

LTOS stellt eine Vielzahl von Informationen über den aktuellen Zustand des Systems zur Verfügung. Eine ganze Reihe von detaillierten Informationen können über Systembefehle angezeigt werden. Oft ist allerdings die Informationsflut und das Format der Ausgabe dieser Befehle nicht geeignet, die interessanten und wichtigen Werte und Informationen schnell zu erfassen. Daher stellt LTOS V6 eine Reihe von Befehlen zur Verfügung, deren Ausgabe optimiert wurde und die sich auf die wichtigsten Angaben konzentriert.

Der Abruf dieser Informationen erfolgt durch die zwei CLI Befehle „show“ und „monitor“. Dabei zeigt „show“ den aktuellen Zustand an und beendet sich dann, d.h. die CLI Eingabeaufforderung erscheint wieder, während „monitor“ die gewählte Information kontinuierlich anzeigt und somit auch automatisch aktualisiert. Der „monitor“ Befehl muss mittels CTRL+C Tastenkombination abgebrochen werden.

Nach dem „show“ bzw. „monitor“ Befehl wird auf der Befehlszeile angegeben, welche Information man abrufen möchte. Diese verschiedenen Informationen werden von sogenannten Plugins zur Verfügung gestellt und jedes Plugin kann eine bestimmte Art von Information anzeigen. So kann das „ip“ Plugin beispielsweise anzeigen, welche IP Adressen in dem System gerade aktiv sind. Um diese Information anzuzeigen, kann man also entweder „show ip“ angeben (damit wird einmalig die Liste der gerade aktuellen IP Adressen angezeigt) oder „monitor ip“ (damit wird die Liste angezeigt und z.B. alle 10 Sekunden aktualisiert, bis man CTRL+C drückt).

Im Folgenden werden die verfügbaren Plugins mit ihren ggf. weiteren Parametern vorgestellt.

3.7.1 cpuload - Anzeige der Prozessorauslastung

Zweck

Das „cpuload“ Plugin zeigt die momentane Auslastung der CPUs an.

Aufruf und Parameter

 # show cpuload

oder

 # monitor cpuload

Der Befehl kennt keine Parameter. Folgende Zeile wird ausgegeben:

```
Tue Jan 21 12:52:26 UTC 2014 Cpu(s): 3.0%us, 4.8%sy, 0.0%ni, 92.1%id,
0.0%wa, 0.0%hi, 0.1%si, 0.0%st Loadavg: 0.36 0.21 0.19 1/80 12803
```

Wobei „Tue Jan 21 12:52:26 UTC 2014“ dem aktuellen Datum/Uhrzeit entspricht, „Cpu(s): 3.0%us, 4.8%sy, 0.0%ni, 92.1%id, 0.0%wa, 0.0%hi, 0.1%si, 0.0%st“ die prozentuale Auslastung der CPUs des Systems angibt (us=User, sy=System, ni=nice, wa=I/O wait, hi=Hardware IRQ, si=Software IRQ, st=Steal Time), „Loadavg: 0.36 0.21 0.19“ die Loadaverage Werte für die letze Minute, die letzten 5 Minuten und die letzten 10 Minuten sowie „1/80 12803“ die Anzahl der Laufenden/Gesamten Prozesse und die letzte verwendete Prozess-ID.

Im „monitor“ Modus wird diese Zeile ca. einmal alle 5 Sekunden ausgegeben, bis CTRL+C gedrückt wird.

Beispiele

 # show cpuload
(zeigt die aktuelle CPU Auslastung an)

 # monitor cpuload
(Zeigt die aktuelle CPU Auslastung kontinuierlich alle 5 Sekunden an)

 # s c
(Kurzform von „show cpuload“)

```
# m c
```
(Kurzform von „monitor cpuload“)

3.7.2 devices - Auflistung von Meinberg Modulen und Baugruppen

Zweck

Mit dem „devices“ Plugin können alle im System erkannten Meinberg Hardware-Bestandteile aufgelistet werden.

Aufruf und Parameter

 # show devices

oder

 # monitor devices

Im „monitor“ Modus wird die Liste alle 10 Sekunden erneut ausgegeben, bis CTRL+C gedrückt wird.

Die Ausgabe sieht zum Beispiel folgendermaßen aus:

```
System Details
System ID: M200
Backplane: P
CPU Carrier: V33
Platform: AMDCONGA
CPU Board: E900
CPU ID: CPU=AuthenticAMD CPUID=AuthenticAMD MODELID=...
RAM: 100868 kB

Found 1 reference clock[s]
GPS170 :2.29 S/N: 11123120 BinaryPort:2 TimeStrPort:0

System Components
Bus/Id Device Product Ver Serial Status
USB 001/005: 1938:0101 Meinberg CPC - Control Panel Controller 1.12 1.0.0
0x0001
```

Beispiele

```
# show devices
```
(zeigt die System Details und die aktuelle Liste der erkannten Meinberg Hardwarekomponenten an)

```
# monitor devices
```
(wie bei „show devices“, allerdings wird die Anzeige alle 10s automatisch aktualisiert)

```
# s d
```
(Kurzform von „show devices“)

```
# m d
```
(Kurzform von „monitor devices“)

3.7.3 ip - Anzeige der IPv4 und IPv6 Adressen

Zweck

Mit dem „ip“ Plugin können alle aktiven IP Adressen angezeigt werden.

Aufruf und Parameter

show ip ***[Filter]***

oder

monitor ip [Filter]

Der „[Filter]“ Parameter ist optional und kann dazu verwendet werden, nur bestimmte IP Adressen anzuzeigen. Er funktioniert wie ein Suchbegriff.

Im „monitor“ Modus wird die Liste alle 10 Sekunden erneut ausgegeben, bis CTRL+C gedrückt wird.

Die Ausgabe von „show ip“ bzw. „monitor ip“ sieht z.B. so aus:

```
Currently Active Network Interfaces:

lan0:0 linklocal ipv6 fe80::213:95ff:fe0a:580b/64
lan0.120 static ipv4 172.16.25.200/255.255.000.000
lan0:0 static ipv6 bad:babe:25::200/64
lan1:1 dhcp ipv4 192.168.10.12/255.255.255.000
```

Die erste Spalte ist der Interface Name, der sich aus dem Bezeichner des physikalischen Netzwerkports (z.B. „lan0“ für den ersten Port) und entweder der ID des virtuellen Interfaces (:0 für das erste logische Interface) oder, bei VLAN Interfaces, der VLAN-ID des virtuellen Interfaces zusammensetzt (z.B. „.120“ für VLAN ID 120).

Die zweite Spalte gibt den Typ der Adresse an, also entweder „static“ für statisch konfigurierte IP Adressen, „dhcp“ für per DHCP/DHCPv6 zugewiesene Adressen, „linklocal“ für die IPv6 Linklocal Adresse des Interfaces oder „ra“ für per Router Advertiser zugewiesene IPv6 Adressen.

In der dritten Spalte wird angegeben, ob es sich um eine IPv4 oder IPv6 Adresse handelt.

Die vierte und letzte Spalte gibt schließlich die IP Adresse und Netzwerkmaske bzw. -Prefixlänge an.

Beispiele

show ip
(zeigt die die aktuelle Liste aller aktiven IP Adressen an)

monitor ip
(wie bei „show ip“, allerdings wird die Anzeige alle 10s automatisch aktualisiert)

show ip lan0
(zeigt alle IP Adressen an, die dem physikalischen Port lan0 zugeordnet sind)

show ip ipv6
(zeigt nur alle IPv6 Adressen an)

show ip dhcp
(zeigt alle per DHCP oder DHCPv6 zugewiesenen Adressen)

 # monitor ip bond

(Zeigt eine alle 10s automatisch aktualisierte Liste der IP Adressen aller Interfaces an, die Teil einer Bonding Gruppe sind)

3.7.4 lantimelog - LANTIME Protokolldatei anzeigen

Zweck

Das „lantimelog“ Plugin für den „show“ Befehl zeigt die LANTIME Protokolldatei (/var/log/lantime_messages) an. Diese enthält nur die wichtigsten systemrelevanten Protokolleinträge, im Gegensatz zur System-Protokolldatei.

Aufruf und Parameter

 # show lantimelog [**Filter**]

oder

 # monitor lantimelog [**Filter**]

Im „monitor“ Modus wird das Ende der Protokolldatei angezeigt und dann alle weiteren neu erzeugten Einträge, solange bis CTRL+C gedrückt wird. Wird ein Filter angegeben, so werden nur die Protokolleinträge angezeigt, die den angegebenen Text enthalten. Wird kein Filter angegeben, werden alle Protokolleinträge aufgelistet.

Die Ausgabe sieht zum Beispiel folgendermaßen aus:

```
# show lantimelog
2014-11-20 13:26:55 UTC: LANTIME -> OSCILLATOR ADJUSTED [Refclock: 1 ]
2014-11-20 13:26:07 UTC: LANTIME -> NORMAL OPERATION
2014-11-20 13:26:03 UTC: LANTIME -> NETWORK LINK UP [Affected LAN
Interface: 1 ]
2014-11-20 13:25:57 UTC: LANTIME -> NTP RESTART
2014-11-20 13:25:57 UTC: LANTIME -> NTP SYNC TO GPS
#
```

Beispiele

 # show lantimelog

(zeigt das gesamte LANTIME Protokoll an)

 # show lantimelog ntp

(zeigt alle LANTIME Protokolleinträge an, die das Wort „NTP“ enthalten)

 # monitor lantimelog

(zeigt die letzten 20 Einträge an und wartet dann auf neue Einträge, die sofort angezeigt werden - Abbruch mit STRG+C)

 # s la

(Kurzform von „show lantimelog“)

 # m la

(Kurzform von „monitor lantimelog“)

3.7.5 linkstate - Anzeige des Verbindungsstatus von Netzwerkschnittstellen

Zweck

Das „linkstate“ Plugin kann anzeigen, ob eine physikalische Netzwerkschnittstelle verbunden ist und wenn ja, mit welcher Geschwindigkeit und welchem Duplex-Modus sie läuft.

Aufruf und Parameter

show linkstate ***[Filter]***

oder

monitor linkstate ***[Filter]***

Der „[Filter]“ Parameter ist optional und kann dazu verwendet werden, nur bestimmte Interfaces anzuzeigen. Er funktioniert wie ein Suchbegriff.

Im „monitor“ Modus wird die Anzeige alle 10 Sekunden erneut aufgefrischt, bis CTRL+C gedrückt wird.

Die Ausgabe von „show linkstate“ bzw. „monitor linkstate“ sieht z.B. so aus:

```
Current LINK state:...
lan0: [00:13:95:12:65:36] 100FDX
lan1: [ec:46:70:ef:3f:e8] NO_LINK
lan2: [ec:46:70:ef:3f:e9] 1000FDX
lan3: [ec:46:70:ef:3f:ea] NO_LINK
```

Die erste Spalte ist der Interface Name (z.B. „lan0“ für den ersten Port).

Die zweite Spalte gibt die MAC Adresse an und in der dritten Spalte steht der aktuelle Verbindungsstatus. Dieser kann entweder „NO_LINK“ sein, d.h. es wurde keine aktive Netzwerkverbindung erkannt, oder es wird die Verbindungsgeschwindigkeit (10, 100, 1000 oder 10000) in MBit/s angegeben sowie der Duplexmodus (FDX=Full Duplex oder HDX=Halbduplex).

Beispiele

show linkstate
(zeigt den Status aller physikalischen Netzwerkschnittstellen an)

monitor linkstate
(wie bei „show linkstate“, allerdings wird die Anzeige alle 10s automatisch aktualisiert)

show linkstate lan0
(zeigt nur den Status des Netzwerkports lan0 an)

s li
(Kurzform von „show linkstate“)

m li
(Kurzform von „monitor linkstate“)

3.7.6 modules - Auflistung der geladenen Treiber bzw. Kernel Module

Zweck

Durch das „modules“ Plugin kann man eine Liste aller gerade geladenen Treiber und sonstige Kernelmodule anzeigen lassen.

Aufruf und Parameter

 # show modules

oder

 # monitor modules

Im „monitor“ Modus wird die Liste alle 10 Sekunden erneut ausgegeben, bis CTRL+C gedrückt wird.

Die Ausgabe sieht zum Beispiel folgendermaßen aus:

```
Loaded kernel modules:
Module Size Used by
ip6table_filter 708 0
ip6_tables 8129 1 ip6table_filter
usb_storage 31278 0
rndis_host 3875 0
cdc_subset 1165 0
cdc_ether 2996 1 rndis_host
bonding 63801 0
ax88179_178a 10848 0
dmfe 13263 0
xt_state 780 0
8021q 11207 0
ipv6 174754 20
squashfs 16978 1
pata_cs5536 2138 1
ext3 85915 0
mbcache 3228 1 ext3
jbd 28294 1 ext3
libahci 14214 0
cgosdrv 18409 0
mdio_bitbang 1515 0
libphy 13845 1 mdio_bitbang
usbnet 10270 4 rndis_host,cdc_subset,cdc_ether,ax88179_178a
ftdi_sio 25482 2
```

Beispiele

 # show modules
(zeigt die Liste der aktuell geladenen Kernelmodule an)

 # monitor modules
(wie bei „show modules“, allerdings wird die Anzeige alle 10s automatisch aktualisiert)

 # s m
(Kurzform von „show modules“)

 # m m
(Kurzform von „monitor modules“)

3.7.7 network - Anzeige des aktuellen Netzwerkstatus

Zweck

Der Befehl „show network“ zeigt eine Gesamtübersicht der im System aktiven Netzwerkkonfiguration an, inklusive der aktuell zugewiesenen IP Adressen sowie des Verbindungsstatus und ggf. den Zustand einer Bonding-Gruppe.

Aufruf und Parameter

 # show network

Die Ausgabe sieht zum Beispiel folgendermaßen aus:

```
=== Physical Interface lan0 : 00:13:95:12:65:36
Speed: 100Mb/s
Duplex: Full
Link detected: yes

Assigned Virtual Interfaces: [:0]
Active Virtual Interfaces: -------------
lan0:0 static ipv4 172.16.25.204/255.255.000.000

=== Physical Interface lan1 : ec:46:70:00:3f:e8
Speed: 10Mb/s
Duplex: Half
Link detected: no

Assigned Virtual Interfaces: [:1]
```

Beispiele

```
# show network
```
(zeigt die aktive Netzwerkkonfiguration an)

```
# s n
```
(Kurzform von „show network“)

3.7.8 processes - Liste der aktiven Prozesse

Zweck

Das „processes“ Plugin zeigt alle gerade im System ausgeführten Prozesse an. Es kann in Kombination mit einem Filter-String dazu verwendet werden, die Ausführung eines bestimmten Befehls bzw. einer bestimmten Software-Komponente zu überprüfen.

Aufruf und Parameter

show processes ***[Filter]***

oder

monitor processes [**Filter**]

Der „[Filter]“ Parameter ist optional und kann dazu verwendet werden, bestimmte Prozesse anzuzeigen. Er funktioniert wie ein Suchbegriff und kann somit entweder Teile des Befehlsnamens enthalten oder auch Prozess-IDs. Es wird hierbei nicht zwischen Groß- und Kleinschreibung unterschieden.

Im „monitor“ Modus wird die Liste sekündlich ausgegeben, bis CTRL+C gedrückt wird.

Die Ausgabe von „show processes“ bzw. „monitor processes“ sieht z.B. so aus:

```
PID TTY STAT TIME COMMAND
1 ? Ss 1:23 /sbin/init
2 ? S 0:00 [kthread
3 ? S 16:12 [ksoftirqd/0]
5 ? S< 0:00 [kworker/0:0H]
7 ? S< 0:00 [kworker/u:0H]
8 ? S< 0:00 [khelper]
9 ? S 0:00 [kworker/u:1]
146 ? S 0:00 [bdi-default]
147 ? S< 0:00 [kblock
155 ? S< 0:00 [ata_sff]
162 ? S 0:00 [khub
268 ? S< 0:00 [rpcio
279 ? S 0:00 [kswapd0]
280 ? S 0:00 [fsnotify_mark]
281 ? S< 0:00 [nfsio
282 ? S< 0:00 [crypto]
552 ? S< 0:00 [deferwq]
554 tty4 Ss+ 0:00 /sbin/getty 38400 tty4
556 tty2 Ss+ 0:00 /sbin/getty 115200 tty2
557 tty3 Ss+ 0:00 /sbin/getty 38400 tty3
600 ? S 0:00 cat /proc/kmsg
615 ? S 1694 ? S 0:00 [scsi_eh_0]
1699 ? S 0:00 [scsi_eh_1]
1704 ? S 1:22 [kworker/u:2]
1777 ? S< 0:00 [kworker/0:1H]
1941 ? S< 0:00 [loop0]
4861 ? S 0:01 [kworker/0:2]
6056 ? S< 0:00 [bond0]
6091 ? S< 0:00 [bond1]
6126 ? S< 0:00 [bond2]
6161 ? S< 0:00 [bond3]
6196 ? S< 0:00 [bond4]
7885 ? Ss 2:55 crond
7903 ? Ss 0:00 /sbin/dbus-daemon -config-file=/etc/dbus-1/system.conf
8998 ? S 0:02 [kworker/0:1]
9153 ? S 1:53 ifplugd -M -f -a -b -d 1 -p -q -i lan0
9179 ? S 1:51 ifplugd -M -f -a -b -d 1 -p -q -i lan1
9205 ? S 1:51 ifplugd -M -f -a -b -d 1 -p -q -i lan2
...
```

Die erste Spalte enthält die Prozess-ID, in der zweiten Spalte wird das Terminal angegeben, auf dem der Prozess läuft („?“ deutet hierbei auf einen internen Prozess hin) und in der dritten Spalte wird der aktuelle Prozess-Status angezeigt:

```
D Uninterruptible sleep (usually IO)
R Running or runnable (on run queue)
S Interruptible sleep (waiting for an event to complete)
T Stopped, either by a job control signal or because it is being traced.
X dead (should never be seen)
Z Defunct („zombie“) process, terminated but not reaped by its parent.
```

In der vierten Spalte wird die kumulative CPU Zeit angezeigt, die der Prozess bisher verbraucht hat. Danach steht der Befehlsname, teilweise mit Parametern, der in diesem Prozess ausgeführt wird.

Beispiele

```
# show processes
```

(zeigt die die aktuelle Liste aller Prozesse an)

```
# monitor processes
```

(wie bei „show processes“, allerdings wird die Anzeige sekündlich automatisch aktualisiert)

```
# show processes ntp
```

(zeigt alle Prozesse an, in deren Befehlsname die Zeichenkette „ntp“ vorkommt - Groß-/Kleinschreibung wird ignoriert)

```
# s p
```

(Kurzform von „show processes“)

```
# m p ntp
```

(Kurzform von „monitor processes ntp“)

3.7.9 route - Anzeige der IPv4 und IPv6 Routingtabelle

Zweck

Das „route“ Plugin des „show“ Befehls listet alle aktiven Netzwerkrouten sowie etwaige Routing-Regeln auf.

Aufruf und Parameter

show route ***[Filter]***

oder

monitor route [Filter]

Der „[Filter]“ Parameter ist optional und kann dazu verwendet werden, nur bestimmte Routing-Einträge anzuzeigen. Er funktioniert wie ein Suchbegriff.

Im „monitor“ Modus wird die Liste alle 10 Sekunden erneut ausgegeben, bis CTRL+C gedrückt wird.

Die Ausgabe von „show route“ bzw. „monitor route“ sieht z.B. so aus:

```
Routing Table Entries:
TABLE DEV TARGET
main lan0 default
main lan0 172.16.0.0/16 proto kernel scope link src 172.16.25.204
local lo broadcast 127.0.0.0 proto kernel scope link src 127.0.0.1
local lo local 127.0.0.0/8 proto kernel scope host src 127.0.0.1
local lo local 127.0.0.1 proto kernel scope host src 127.0.0.1
local lo broadcast 127.255.255.255 proto kernel scope link src 127.0.0.1
local lan0 broadcast 172.16.0.0 proto kernel scope link src 172.16.25.204
local lan0 local 172.16.25.204 proto kernel scope host src 172.16.25.204
local lan0 broadcast 172.16.255.255 proto kernel scope link src
172.16.25.204
main lo local ::1 proto none metric 0
0 lo unreachable default proto kernel metric -1 error -101

Routing Rules:
0: from all lookup local
32766: from all lookup main
32767: from all lookup default
```

Beispiele

```
# show route
```
(zeigt die die aktuelle Liste aller aktiven Routen und Regeln an)

```
# monitor route
```
(wie bei „show route“, allerdings wird die Anzeige alle 10s automatisch aktualisiert)

```
# show route lan0
```
(zeigt alle Routing-Einträge an, die dem physikalischen Port lan0 zugeordnet sind)

```
# s r
```
(Kurzform von „show route“)

```
# m r
```
(Kurzform von „monitor route“)

3.7.10 syslog - System Protokolldatei anzeigen

Zweck

Mit dem „syslog“ Plugin wird die System Protokolldatei (/var/log/messages) angezeigt. Diese enthält alle Fehlermeldungen und Statusupdates für das System.

Aufruf und Parameter

show syslog [**Filter**]

oder

monitor syslog [**Filter**]

Im „monitor“ Modus wird das Ende der Protokolldatei angezeigt und dann alle weiteren neu erzeugten Einträge, solange bis CTRL+C gedrückt wird. Wird ein Filter angegeben, so werden nur die Protokolleinträge angezeigt, die den angegebenen Text enthalten. Wird kein Filter angegeben, werden alle Protokolleinträge aufgelistet.

Die Ausgabe sieht zum Beispiel folgendermaßen aus:

```
# show syslog
Dec 15 10:41:08 timeserver root: Restarting syslog due to configuration
change ...
Dec 15 10:41:08 timeserver syslog-ng[7864]: Termination requested via
signal, terminating;
Dec 15 10:41:08 timeserver syslog-ng[7864]: syslog-ng shutting down;
version=‘2.0.9‘
Dec 15 10:41:08 test_tr0_lt04 syslog-ng[22289]: syslog-ng starting up;
version=‘2.0.9‘
Dec 15 11:19:40 test_tr0_lt04 sshd[5061]: Accepted password for root from
172.16.3.120 port 41449 ssh2
Dec 15 11:19:40 test_tr0_lt04 sshd[5061]: pam_unix(sshd:session): session
opened for user root by (uid=0)
Dec 15 11:19:46 test_tr0_lt04 sshd[5061]: Received disconnect from
172.16.3.120: 11: PECL/ssh2 (http://pecl.php.net/packages/ssh2)
Dec 15 11:19:46 test_tr0_lt04 sshd[5061]: pam_unix(sshd:session): session
closed for user root
#
```

Beispiele

```
# show syslog
```
(zeigt das gesamte System-Protokoll an)

```
# show syslog failed
```
(zeigt alle System-Protokolleinträge an, die das Wort „failed“ enthalten)

```
# monitor syslog
```
(zeigt die letzten 20 Einträge an und wartet dann auf neue Einträge, die sofort angezeigt werden - Abbruch mit STRG+C)

```
# s s
```
(Kurzform von „show syslog“)

```
# m s
```
(Kurzform von „monitor syslog“)

```
# s
```
(Kurzform von „s s“)

```
# m
```
(Kurzform von „m s“)

3.7.11 version - Anzeige der Firmware Version

Zweck

Mit dem „version“ Plugin wird die Versionsnummer der gerade laufenden Firmware angezeigt.

Aufruf und Parameter

 # show version

Die Ausgabe sieht zum Beispiel folgendermaßen aus:

```
Running LTOS V6.16.005 [standar
System Version : Linux test_tr0_lt04 3.7.1 #16 Wed Jul 16 10:33:54 UTC 2014
i586 unknown
```

Beispiele

```
# show version
```
(zeigt die Firmware Version an)

```
# s v
```
(Kurzform von „show version“)

3.8 Systembefehle

Die Überwachung von Systemressourcen wie Flash Speicher oder RAM wird durch eine Reihe von CLI Kommandos ermöglicht, die in diesem Abschnitt beschrieben werden. Es handelt sich dabei hauptsächlich um Befehle zur Anzeige von diversen Ressourcen (wie zum Beispiel „freier Hauptspeicher“). Damit können Sie eventuell auftretende Probleme erkennen und diagnostizieren.

3.8.1 reboot - System neu starten

Zweck

Der **reboot** Befehl startet das gesamte System neu. Dabei werden alle Dienste beendet und danach wird ein Reset ausgelöst. Zu beachten ist, dass etwaige ungesicherte Änderungen an der Konfiguration nicht automatisch gespeichert werden, sodass nach dem Neustart die zuletzt gespeicherte Startup-Konfiguration geladen wird.

Es ist möglich, vor dem Auslösen des Neustarts eine gewisse Zeit zu warten. Das kann auch im Hintergrund passieren und abgebrochen werden, d.h. der **reboot** Befehl wartet eine gewisse Zeit ab, während man im Vordergrund weiterarbeitet. Mit diesem Mechanismus kann man z.B. vor Änderungen an der Netzwerkkonfiguration angeben, dass das System in 5 Minuten neu durchstarten soll. Sorgt dann eine Konfigurationsänderung dafür, dass das System nicht mehr erreichbar wird, dann wird durch den Neustart diese ungesicherte letzte Änderung rückgängig gemacht und das Gerät ist wieder erreichbar. Sollte man nach den Konfigurationsänderungen feststellen, dass alles wie erwartet läuft, kann man den im Hintergrund wartenden Reboot-Prozess durch „**reboot** *stop*“ wieder abbrechen.

Der Reboot Befehl informiert alle in SSH/Telnet/Seriellen Konsolen angemeldeten Benutzer darüber, wann er den Neustart ausführt. Dadurch wird ggf. diesen anderen Benutzern die Möglichkeit gegeben, etwaige Änderungen noch zu speichern und sich ordnungsgemäss abzumelden oder den Reboot Vorgang abzubrechen.

Aufruf und Parameter

\# **reboot** [*DELAY*|*stop*]

Wird als Parameter ein Delay angegeben, dann schläft der Reboot-Befehl für die angegebene Zeit bevor er den Reboot tatsächlich auslöst. Wird das Delay als einfacher Zahlenwert x angegeben, dann wird für x Sekunden gewartet. Es ist aber auch möglich, Minuten oder Stunden anzugeben, indem einem Zahlenwert entweder „m“ oder „h“ nachgestellt wird (siehe Beispiele).

Will man in der gleichen SSH/Telnet/seriellen Sitzung weiterarbeiten, während **reboot** im Hintergrund wartet, muss man dem ganzen Befehl das „&“ Zeichen nachstellen.

Ein etwaiger im Hintergrund wartender Reboot Prozess kann beendet werden, indem man anstatt eines Delays *stop* angeben. Damit kann auch ein von einem anderen Benutzer ausgelöster Reboot Vorgang abgebrochen werden, solange die Wartezeit das erlaubt.

Wird kein Parameter angegeben, nimmt **reboot** die Standardwartezeit von 2 Sekunden an.

Beispiele

```
# reboot
```
(startet das System sofort, d.h. nach 2 Sekunden, neu)

```
# reboot 20
```
(startet das System nach 20 Sekunden Wartezeit neu)

```
# reboot 1h
```
(startet das System nach 1 Stunde Wartezeit neu)

```
# reboot 5m &
```
(startet das System nach 5 Minuten Wartezeit neu, wartet aber im Hintergrund)

```
# reboot stop
```

(bricht einen eventuell wartenden Reboot Vorhang ab und verhindert somit den Neustart)

3.8.2 make_noise - Gerät durch Blinken und Piepen identifizieren

Zweck

Der Befehl **make_noise** lässt das Gerät im 2 Sekundentakt piepen, ausserdem blinkt die rote Alarm LED. Es ist damit möglich, ein Gerät in einem unübersichtlichen Raum zu identifizieren. Bei Geräten mit Front Panel Display kann am Gerät selbst der Alarm durch Druck auf die Taste F2 abgeschaltet werden. Ansonsten stoppt man den Befehl durch Druck auf STRG+C.

Aufruf und Parameter

 # make_noise

Dieser Befehl kennt keine Parameter.

Beispiele

```
# make_noise
```

(führt zum Blinken der Alarm LED und zum Piepen, bis entweder am Gerät F2 gedrückt wurde oder durch STRG+C abgebrochen wurde)

4 Texteditoren

Bei Verwendung des Command Line Interfaces ist das manuelle Bearbeiten von Konfigurationsdateien eine wichtige Funktion. Dafür stellt LTOS V6 zwei verschiedene Texteditoren namens **nano** und **edit** zur Verfügung, die unterschiedliche Möglichkeiten und Funktionsumfänge bieten. Welchen der beiden Texteditoren Sie verwenden, können Sie frei wählen. Eventuell ist die Bildschirmdarstellung auf Ihrem Rechner für einen der beiden Editoren besser als für den anderen oder Sie bevorzugen das Bedienungskonzept einer der beiden Alternativen.

Da auf früheren LANTIME OS Versionen der nano Texteditor immer mit dem Befehl

vi *[filename]*

aufgerufen wurde, gibt es unter V6 einen gleichlautenden Befehl. Dieser gibt einen Hinweis aus, und ermöglicht Ihnen dann, einen der beiden verfügbaren Editoren auszuwählen, mit dem die angegebene Datei *[filename]* editiert werden soll.

4.1 nano

Der nano Texteditor ist ein Opensource Texteditor (siehe http://www.nano-editor.org/ für weitere Infos), der klein, schnell und übersichtlich ist. Dieser Editor wurde in früheren LTOS Versionen standardmässig gestartet, wurde dort aber mit dem „vi“ Befehl aufgerufen.

Aufruf und Parameter

 # nano [filename]

ruft nano auf und öffnet die angegebene Datei [filename]. Wird dieser Parameter weggelassen, öffnet sich nano und ermöglicht das Anlegen einer Datei. Beim Verlassen und speichern (siehe unten) wird dann der Dateiname abgefragt.

Bedienung - Wichtige Funktionen

Befehlstasten
Der „nano“-Editor verwendet Tastenbefehle, um die Programmfunktionen auszuwählen. Dabei wird immer die STRG-Taste (CTRL auf englischen Tastaturen) gedrückt gehalten und eine Buchstabentaste angetippt, die dann eine bestimmte Funktion auslöst.

Speichern von geänderten Dateien
Das Speichern einer Datei wird durch STRG+O („WriteOut“) ausgelöst. Dabei wird der Editor nicht verlassen. Nach dem Druck auf STRG+O wird noch mal gefragt, unter welchem Dateinamen die Datei gespeichert werden soll. Drückt man hier einfach ENTER, wird der alte Name verwendet. Mit STRG+C kann man das Speichern abbrechen und zum Editor zurückkehren.

Editor verlassen
Mit STRG+X kann der Editor verlassen werden. Es erscheint ein Dialog, in dem man entweder mit „Y“ (für „Yes“) die Änderungen speichern kann, mit „N“ (für „No“) die Änderungen nicht speichert und trotzdem den Editor verlässt oder mit STRG+C das Verlassen des Editors abbricht und - ohne zu speichern - zurück in den Editor gelangt.

Suchen/Ersetzen
Um einen bestimmten Suchbegriff in der gerade geöffneten Datei zu finden, kann man STRG+W drücken. Um eine bestimmte Zeichenkette durch eine andere zu ersetzen, kann man den entsprechenden Dialog mit STRG+\ aufrufen. Der nano Editor unterstützt die Suche mithilfe von regulären Ausdrücken (Regular Expression).

Weitere Funktionen
Eine Vielzahl von zusätzlichen Befehlen und Funktionen ist über spezielle Tastenkombinationen erreichbar.

Mit STRG+G kann man eine Hilfe aufurfen, die die Funktionen mit ihren zugehörigen Tastenkombinationen auflistet.

4.2 edit

Hinter dem edit Texteditor verbirgt sich der Editor der Midnight Commander Software (mcedit), ein Opensource Projekt (siehe http://www.midnight-commander.org). Er bietet umfangreiche Features, ein Menü sowie farbige Darstellungsoptionen (wenn das vom Terminal unterstützt wird).

Aufruf und Parameter

edit [**filename**]

ruft edit auf und öffnet die angegebene Datei [filename]. Wird dieser Parameter weggelassen, startet der Editor und ermöglicht das Anlegen einer Datei. Beim Verlassen und speichern (siehe unten) wird dann der Dateiname abgefragt.

Bedienung - Wichtige Funktionen

Funktionstasten
Dieser Editor verwendet die Funktionstasten, um die wichtigsten Programmfunktionen aufzurufen. Wenn Ihr Terminalprogramm die Funktionstasten nicht korrekt überträgt oder Sie sie aus einem anderen Grund nicht verwenden können, kann durch Drücken der Escape-Taste („ESC“) und direkt danach Eingabe einer Nummer 1-9 der Druck auf eine der Tasten F1 bis F9 simuliert werden. ESC+0 entspricht F10 und ist wichtig für das Verlassen des Editors (siehe unten).

Speichern von geänderten Dateien
Das Speichern einer Datei wird durch F2 ausgelöst. Dabei wird der Editor nicht verlassen. Nach dem Druck auf F2 (oder ESC+2) wird noch mal gefragt, ob tatsächlich gespeichert werden soll. Man kann dann mit „S“ (für „Save“) speichern oder das Speichern abbrechen mit „C“ (Cancel).

Editor verlassen
Mit F10 (oder ESC+0) kann der Editor verlassen werden. Wurde die Datei geändert und die Änderungen noch nicht gespeichert, erscheint ein Dialog, in dem man entweder mit „Y“ (für „Yes“) die Änderungen speichert, mit „N“ (für „No“) die Änderungen nicht speichert und trotzdem den Editor verlässt oder mit „C“ (für „Cancel“) das Verlassen des Editors abbricht und - ohne zu speichern - zurück in den Editor gelangt.

Suchen/Ersetzen
Um einen bestimmten Suchbegriff in der gerade geöffneten Datei zu finden, kann man die F7 Taste (oder ESC+7) drücken. Um eine bestimmte Zeichenkette durch eine andere zu ersetzen, kann man den entsprechenden Dialog mit F4 (ESC+4) aufrufen. Dabei sind eine Vielzahl von Optionen auswählbar. Der mcedit Editor unterstützt die Suche mithilfe von regulären Ausdrücken (Regular Expression), beim Ersetzen kann man durch Auswahl der „Prompt on Replace“ Funktion angeben, dass man vor jedem Ersetzen noch mal gefragt werden möchte. Die Option „replace All“ ist wichtig, wenn man viele bzw. alle Vorkommen des Suchbegriffes ersetzen möchte.

Weitere Funktionen
Eine Vielzahl von zusätzlichen Befehlen und Funktionen ist über das Menü des mcedit erreichbar, das man mit F9 (ESC+9) erreichen kann. Sie können mit den Pfeiltasten im Menü navigieren und entweder mit ENTER einen Menüpunkt auswählen oder mit ESC das Menü verlassen und zur bearbeiteten Datei zurückzukehren.

4.3 vim

Der VImproved Texteditor („vim“) ist ein Opensource Texteditor (siehe http://www.vim.org/ für weitere Infos), der eine sehr grosse Anzahl an Funktionen und Features bietet, allerdings relativ komplex zu bedienen ist und daher nicht für Einsteiger empfohlen wird.

Aufruf und Parameter

 # vim [filename]

ruft vim auf und öffnet die angegebene Datei [filename]. Wird dieser Parameter weggelassen, öffnet sich vim mit einer leeren und unbenannten Datei. Die Hilfe-Texte für vim sind unter LTOS6 aus Platzgründen nicht verfügbar. Bitte greifen Sie auf die Dokumentation der Website www.vim.org zurück.

Bedienung - Wichtige Funktionen

Editor-Befehle und Betriebsmodus
Der vim Editor hat drei verschiedene Modi, in denen unterschiedliche Funktionen und Befehle ausgeführt werden können. Nach dem Start befindet man sich im sogenannten „Normal Mode“. In diesem Modus können Funktionen durch das Drücken von Buchstabentasten direkt ausgeführt werden. Nach Druck auf die Doppelpunkt-Taste „:“ landet man im sogenannten Kommandomodus. Den Einfüge/Bearbeiten-Modus erreicht man im Normal Mode durch Eingabe eines „i“ oder Drücken der Einfüge-Taste (Einfg). Durch mehrfaches Drücken der Escape-Taste (ESC) gelangt man zurück in den Normalmodus.

Speichern von geänderten Dateien
Das Speichern einer Datei wird im Kommandomodus erledigt. Vom Normal Mode aus gelangt man mit „:“ in den Kommandomodus, dort kann durch Eingabe eines „w“ und nachfolgendem ENTER der Speichervorgang ausgeführt werden.

Editor verlassen
Zum Verlassen des Editors kann man im Kommandomodus den Befehl „q“ verwenden. Wurde die Datei geändert und noch nicht gespeichert, muß man entweder anstelle des „q“ Befehls den „wq“ BEfehl ausführen (Speichern und beenden) oder kann den Editor mit dem Befehl „q!“ beenden, ohne zu speichern. Im „Normal Mode“ ist es auch möglich, durch zweimaliges Drücken der „z“ Taste zu speichern und danach den Editor zu verlassen.

Weitere Funktionen
Der vim Editor hat einen sehr grossen Funktionsumfang. Mehr zu den vielen Funktionen können Sie im Internet lesen. Das frei erhältliche PDF Buch „The Vim Tutorial and Reference“ von Steve Oualline hat einen Umfang von 800 Seiten.