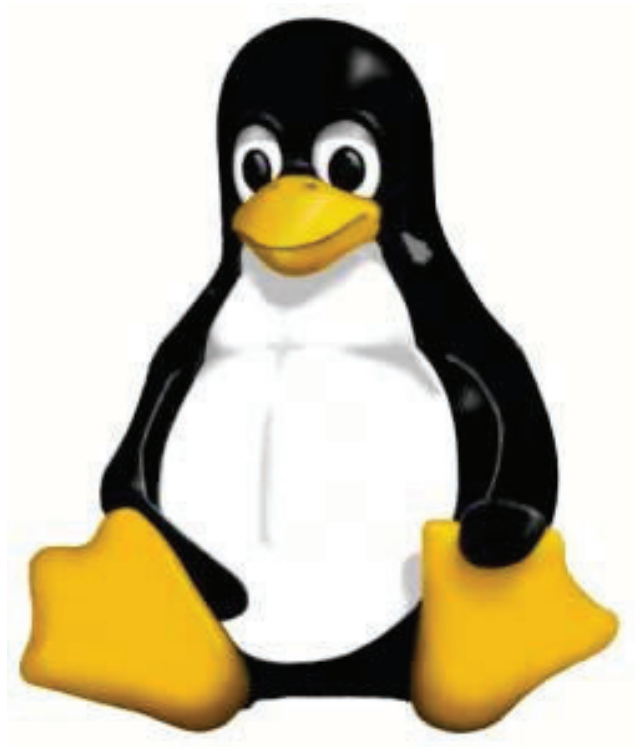


Linux-Crash-Kurs

Leitprogramm



Autor: Beat Loosli
Datum: September 2006 / August 2007 / August 2010

Inhaltsverzeichnis

Linux Crash-Kurs	3
Einführung in das Thema	3
Arbeitsanleitung	4
Kapitel 1 - Die Bash, Turbo auf der Konsole	5
Kapitel 2 – Wie werden unter Linux die Daten verwaltet	9
Kapitel 3 – <code>find</code> und <code>grep</code> – Die Sherlock Holmes	15
Kapitel 4 – Einpacken, auspacken, wegpacken mit <code>tar</code>	19
Kapitel 5 – Informationen zu meinem System	22
Lösungen	26

Linux Crash-Kurs

Einführung in das Thema

Im Serverbereich hat Linux seit längerer seinen Stammplatz gefunden. Immer mehr Firmen setzen ihre Server für Web-, Mail- und Datenbankdienste auf dem Betriebssystem Linux auf. Linux stellt jedoch für windowsgewohnte Benutzerinnen und Benutzer eine relative hohe Einstiegshürde dar. Warum ist dies eigentlich so? Auch Linux tritt heute mit farbigen und intuitiv bedienbaren Desktopoberflächen auf. Um es gerade vorwegzunehmen: Ein Serverdienst unter Linux benötigt keine grafische Oberfläche! Ausser Verschleiss von Systemressourcen und potentiellen Sicherheitslücken bieten diese Oberflächen keinen erheblichen Nutzen. Die Stärke einer Serveradministration unter Linux liegt auf der Kommandozeilebene.

Was sollten Sie mitbringen?

Etwas Freude und ein wenig Forschungsdrang sollten eigentlich genügen um diesen Crash-Kurs erfolgreich zu absolvieren.

Was lernen Sie hier?

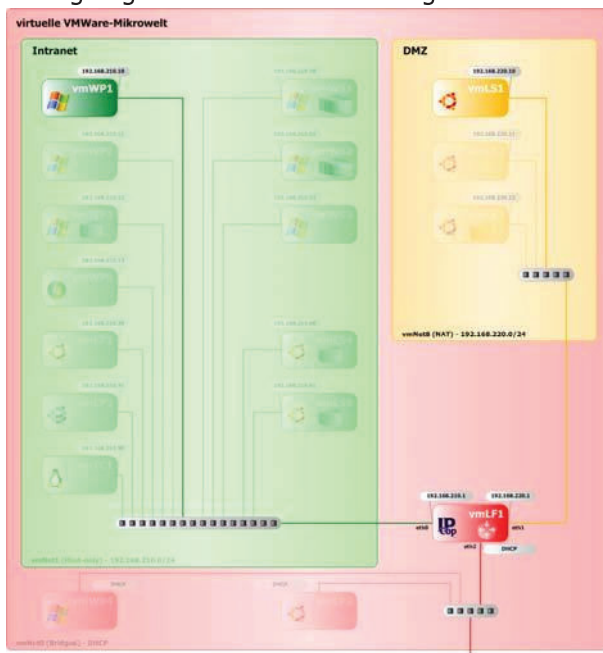
Das Arbeiten auf der Kommandozeilebene sollte Ihnen nach diesem Kurs keine Mühe mehr machen. Zudem werden Sie einen guten Einblick in die Tiefen von Linux gewinnen und Sie werden in der Lage sein, die verschiedenen Serverdienste via Konsole zu installieren und zu administrieren.

Was Sie nicht lernen!

Wir werden die Möglichkeiten einer grafischen Oberfläche unter Linux wie die KDE oder Gnome nicht benutzen. Diese Oberflächen sind intuitiv erlernbar und wenn Sie unter Windows Erfahrung haben, werden Sie die meisten Funktionen selber herausfinden.

Welche Hilfsmittel benötigen Sie?

Als Nachschlagewerk können Sie zusätzlich „Die Linuxfibel“ auf folgender Website benutzen: <http://de.linwiki.org/wiki/Linuxfibel>. Die Maschinen *vmLF1* und *vmLS1* müssen für die Übungen gestartet sein. Als Arbeitsgerät wird die Linux-Maschine *vmLS1* eingesetzt.



Arbeitsumgebung gibbix.ch



Loginschirm vmLS1
(User: vmadmin - Passwort: gibbiX12345)

Arbeitsanleitung

Mit Hilfe dieses Leitprogramms können Sie sich selbständig mit der Arbeitsweise auf einer Linux-Konsole bekannt machen.

```
vmadmin@umlS1:~$ ls -all /etc/*conf
-rw-r--r-- 1 root root 2986 2009-05-18 00:56 /etc/adduser.conf
-rw-r--r-- 1 root root 6282 2009-05-18 00:56 /etc/ca-certificates.conf
-rw-r--r-- 1 root root 2969 2009-03-24 06:30 /etc/debconf.conf
-rw-r--r-- 1 root root 600 2009-03-27 23:43 /etc/deluser.conf
-rw-r--r-- 1 root root 34 2008-02-19 05:33 /etc/e2fsck.conf
-rw-r--r-- 1 root root 2689 2008-03-26 18:44 /etc/gai.conf
-rw-r--r-- 1 root root 92 2008-12-23 19:53 /etc/host.conf
-rw-r--r-- 1 root root 240 2009-05-18 00:58 /etc/kernel-img.conf
-rw-r--r-- 1 root root 34 2009-05-18 00:55 /etc/ld.so.conf
-rw-r--r-- 1 root root 803 2009-02-12 16:37 /etc/mke2fs.conf
-rw-r--r-- 1 root root 475 2006-08-28 18:33 /etc/nsswitch.conf
-rw-r--r-- 1 root root 552 2009-03-21 10:28 /etc/pam.conf
-rw-r--r-- 1 root root 73 2009-06-26 11:16 /etc/resolv.conf
-rw-r--r-- 1 root root 85602 2008-07-04 10:09 /etc/sensors.conf
-rw-r--r-- 1 root root 2277 2009-03-18 23:17 /etc/sysctl.conf
-rw-r--r-- 1 root root 1626 2009-01-23 19:33 /etc/syslog.conf
-rw-r--r-- 1 root root 1260 2008-05-30 08:22 /etc/ucf.conf
vmadmin@umlS1:~$ _
```

typische Linux-Konsole

Das Programm besteht aus verschiedenen Kapiteln. Diese sind immer gleich strukturiert:



Lernziele

Was Sie können sollten nach dem Bearbeiten des Kapitels.



Übersicht

Worum geht es in diesem Kapitel.



Theorie

Theoretische Teile und Erklärungen.

Beispiel

Beispiele

Beispiele zum Ausprobieren und Nachvollziehen.



Lernkontrollen

Selbständige Aufgaben.



Hinweise

Wichtige Ergänzungen zum Thema.



Kapiteltest

Lösen Sie an dieser Stelle die Aufgaben selbständig. Melden Sie sich bei der Lehrperson, um Ihre Lösungen zu besprechen. Zusammen mit der Lehrperson wird entschieden, ob Sie mit dem nächsten Kapitel weiterfahren können oder ob das aktuelle Kapitel noch einmal wiederholt werden soll.

Kapitel 1 - Die Bash, Turbo auf der Konsole



Lernziele

Nach diesem Kapitel kennen Sie die Grundlagen um die *Bash* effizient zu bedienen:

- Sie verstehen die Kommandoeingabe und den Aufbau der Befehlssyntax
- Sie können die internen Hilfestellungen für die Befehle benutzen
- Sie kennen den Zweck der wichtigsten Tastenkürzel
- Sie wissen, wie sich die *Bash* schon benutzte Befehle merken kann
- Sie können die Ein- und Ausgabeumleitungen anwenden



Übersicht

Die *Bash* ist eine beliebte Shell unter Linux. Für Linux ist die *Bash* in der Regel das erste Programm, das nach dem erfolgreichen Anmelden eines Benutzers gestartet wird. Für uns wird die *Bash* die zentrale Schnittstelle zu unserer Linux-Maschine sein.

Siehe auch Linux-Fibel: Linuxfibel - Die Bash - Eingabehilfen



Die unter Linux verwendeten Kommandointerpreter werden Shell genannt. Als Kommandointerpreter ist die Shell ein wichtiges Bindeglied zwischen Anwender und Betriebssystem. Es gibt verschiedene Shells für Linux wie zum Beispiel die *cs*h (C-Shell), die *ks*h (Korn-Shell) oder die *Bash* (Bourne Again Shell). Die *Bash* ist eine der leistungsfähigsten Benutzerschnittstellen unter Linux.

Vom Sprachumfang her sind alle üblichen Shells als vollwertige Skriptsprachen zur Programmierung und zur Automatisierung von Aufgaben verwendbar; die Abgrenzung zu reinen Skriptsprachen (z. B. Perl, AWK) besteht darin, dass Shells besondere Mittel für den interaktiven Dialog mit dem Anwender bereitstellen, die vom Ausgeben eines Prompts im einfachsten Fall bis hin zur Möglichkeit des Editierens der eingegebenen Befehle oder zur Jobsteuerung reichen.

Kommandoeingabe

In der Shell können Sie die gewünschten Befehle direkt hinter dem Promptzeichen eingeben. Die Befehle haben in der Regel folgenden Aufbau:

Befehlsname [-Option] [Argumente]

Beispiel

```
vmadmin@vmLS1:~> cp -r ./Documents ./Archiv
```

Prompt

Befehl:
kopieren

Option:
alle Daten inkl. Unterverzeichnisse

Argumente:
Quelle, Ziel



Optionen werden fast immer mit einem Minus (-) eingeleitet. Für viele Optionen gibt es eine Langform, die mit zwei Minuszeichen (--) eingeleitet werden. Mehrere Optionen in der Kurzform müssen jeweils nicht mit einem Leerzeichen getrennt werden. Sie können direkt hintereinander angegeben werden.

Die Tilde (~) im Prompt weist übrigens darauf hin, dass man sich im Homeverzeichnis befindet, in unserem Fall im Homeverzeichnis des Benutzers *vmadmin*.

Beispiel

Sie möchten mit dem Befehl *ls* den Inhalt des aktuellen Verzeichnisses auflisten und dies mit den Optionen *-l* (formatierte Ausgabe) und *-a* (versteckte Dateien anzeigen).

```
vmadmin@vmLS1:~> ls -al
```

Hilfe zu den Befehlen

Während dieses Crash-Kurses werden Sie etliche Linux-Befehle kennen lernen. Es ist jedoch nicht möglich, sich die zum Teil umfangreichen Optionslisten auswendig zu merken. Eine Übersicht die zu einem Befehl verfügbaren Optionen, können Sie mit dem Befehl `man <Befehl>` oder `<befehl> --help` aufrufen.

Beispiel

Um zum Beispiel die möglichen Optionen für den Befehl `ls` aufzulisten, können Sie eine der zwei folgenden Möglichkeiten benutzen:

Möglichkeit 1

```
vmadmin@vmLS1:~> man ls
```

Um die `man`-Page zu verlassen, drücken Sie die Taste `Q`.

Möglichkeit 2

```
vmadmin@vmLS1:~> ls --help
```



Probieren Sie die Möglichkeiten direkt an Ihrer Konsole aus indem Sie die Hilfe beziehungsweise die `man`-Page von den Befehlen `cp`, `rm`, und `ls` studieren.

Tastenkürzel

Wenn Sie auf der Konsole ein Virtuose werden wollen, sollten Sie sich mit den zahlreichen Hilfsmitteln auskennen, welche die *Bash* zur Verfügung stellt.

Die aktuelle Befehlszeile lässt sich wie in einem Textbearbeitungsprogramm editieren. Es ist möglich einzelne Zeichen zu löschen, zu ersetzen, an beliebiger Stelle hinzuzufügen. Die wichtigsten Tasten für die Arbeit auf der Konsole:



Cursor auf der aktuellen Befehlszeile vor und zurückbewegen



wechselt durch die zuletzt eingegebenen Befehle



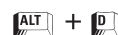
vervollständigt den Befehl, Datei- oder Verzeichniseingaben, sobald sie eindeutig sind. Zweimal drücken, ruft gegebenenfalls alle vorhandenen Möglichkeiten auf. **Super Taste!!**



Zeichen rückwärts löschen / Zeichen vorwärts löschen



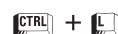
Cursor zum Beginn der Zeile / Cursor an das Ende der Zeile



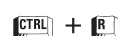
Wort löschen, Cursor muss auf dem ersten Zeichen stehen



ab dem Cursor bis zum Ende der Zeile löschen



löscht den gesamten Bildschirm (wie Befehl `clear`)



ruft die Suchfunktion der *Bash* auf. Wenn das erste Zeichen des Befehls eingegeben wird, sucht die *Bash* nach alten Eingaben, welche die entsprechenden Zeichen enthalten.



Geben Sie eine Zeile mit beliebigem Text ein. Testen Sie jetzt der Reihe nach alle oben genannten Tastenkürzel, bis Sie das Gefühl dafür bekommen haben. Testen Sie insbesondere auch die Suchfunktion **Ctrl + R**, sie ist etwas gewöhnungsbedürftig, aber sehr nützlich. Daneben sollten Sie auch die **Tab**-Funktion ausgiebig erkunden – sie ist eine der mächtigsten Funktionen der *Bash*, da sich die Eingaben um ein Vielfaches beschleunigen lassen.

Die Bash-History

Vielleicht haben Sie sich schon gefragt, woher die *Bash* weiss, welche Eingaben Sie vorher schon einmal gemacht haben. Schliesslich merkt sich die *Bash* auch alle Eingaben, die Sie gestern, vorgestern oder letzte Woche gemacht haben.



Schauen Sie sich mit dem Befehl `less` oder `more` die Datei `~/.bash_history` in Ihrem Homeverzeichnis an. Sie werden sehen, dass in dieser Datei sämtliche Befehle von früher gespeichert sind.

```
vmadmin@vmLS1:~> less ~/.bash_history
```

Siehe auch [Linux-Fibel: Linuxfibel - Die Bash – Eingabehilfen - Das Gedächtnis ...](#)

Ein- und Ausgabeumleitungen

Sie können Ausgaben eines Befehls in verschiedener Weise weiterverarbeiten und Befehle kombinieren. Ohne weitere Angabe listet der Befehl `ls -al` den formatierten Inhalt des aktuellen Verzeichnisses am Bildschirm auf. In gewissen Fällen wäre es praktisch, wenn die Ausgabe in eine Datei geschrieben würde.

Beispiel

Probieren Sie folgenden Befehl aus.

```
vmadmin@vmLS1:~> ls -al > inhalt.txt
```

Damit wird die Ausgabe des Befehls in die Datei namens `inhalt.txt` geschrieben. Dazu wird die schliessende Spitzklammer (`>`) verwendet. Die Datei muss nicht existieren – gegebenenfalls wird sie neu erstellt. Wenn sie schon vorhanden ist wird der alte Inhalt überschrieben. Möchten Sie den neuen Inhalt an eine bestehende Datei anhängen, müssen Sie zwei Spitzklammern verwenden.

```
vmadmin@vmLS1:~> ls -al >> inhalt.txt
```

Sie können sich nun den Inhalt der Datei mit dem Befehl `less` anschauen.











```
vmadmin@vmLS1:~> less inhalt.txt
```



Endungen wie `.txt` sind unter Linux nicht notwendig, um den Dateityp zu kennzeichnen. Sie können jedoch zu einer besseren Übersicht beitragen.

Beispiel

Möchten Sie Daten aus einer Datei auslesen und weiterverarbeiten, müssen Sie die öffnende Spitzklammer (`<`) benutzen. Um dies besser zu verstehen, führen Sie folgendes Beispiel durch und ziehen Sie ihre eigenen Schlüsse.

```
vmadmin@vmLS1:~> cat > unsortiert.txt   
tango   
bravo   
november   
charlie   
zulu   
sierra   
 +   
vmadmin@vmLS1:~> sort < unsortiert.txt 
```



Mit der vorhergehenden Übung konnten Sie zusätzlich lernen, wie man auf eine sehr einfache Art eine neue Datei erstellt.

Pipes

Oft wird es vorkommen, dass Sie die Ausgabe eines Befehls in einem zweiten Kommando weiterverwenden möchten. Lassen Sie sich einmal mit `ls -l` den Inhalt des Verzeichnisses `/dev` anzeigen. Das Ergebnis ist eine Liste mit hunderten von Einträgen. Mit Hilfe einer Pipe können wir die Ausgabe dem Befehl `less` übergeben und somit bequem in der Liste hin- und herblättern.

Beispiel

```
vmadmin@vmLS1:~> ls -l /dev | less
```



Pipes (engl. Röhren) werden auf der CH-Tastatur mit der Kombination **AltGR + 7** zwischen zwei Befehle eingeschoben. Sie leiten die Ausgabe des ersten Befehls als Eingabe zum zweiten Befehl.



Kapiteltest 1

Bei Aufgaben, die an der Konsole zu lösen sind, notieren Sie als Lösung die korrekte Befehlsfolge.

1. Wie können Sie herausfinden was der der Befehl `rm -i inhalt.txt` bewirkt?

[illegible]

2. Listen Sie das Verzeichnis `/etc` mit dem Befehl `ls -al` in die Datei `etc_inhalt.txt` in ihrem Homeverzeichnis und zeigen Sie diese Datei im Anschluss mit dem Befehl `less` am Bildschirm an.

[illegible]

3. Wie können Sie herausfinden, wie viele Linux-Befehle mit dem Buchstaben b beginnen?

[illegible]

4. Lassen Sie sich Ihre *Bash*-History anzeigen.

[illegible]

Kapitel 2 – Wie werden unter Linux die Daten verwaltet



Lernziele

Nach diesem Kapitel kennen Sie den Aufbau eines Linux-Filesystems und die wichtigsten Befehle um Dateien zu manipulieren:

- Sie verstehen die Hierarchie des Linux-Filesystems
- Sie können innerhalb des Dateisystems navigieren
- Sie können neue Dateisysteme im Verzeichnisbaum einbinden und entfernen
- Sie können Dateien und Verzeichnisse erstellen, umbenennen und löschen



Übersicht

Die Dateiablage unter Linux wird über eine hierarchische Baumstruktur abgebildet. Der Zugang zu allen möglichen Dateisystemen (Festplatte, Netzwerk, CD-ROM, DVD, Diskette) wird über diese Hierarchie ermöglicht. Dateisysteme werden mit dem Befehl `mount` in die Baumstruktur eingehängt, beziehungsweise mit dem Befehl `umount` wieder aus dem Baum entfernt. Den Dateibaum muss man sich umgedreht vorstellen, die Wurzel (dargestellt durch `/`) befindet sich in der obersten Hierarchie, danach erfolgt die Verzweigung in weitere Ebenen. Die Dateien kann man sich als Blätter und die Verzeichnisse als Äste der Baumstruktur vorstellen. Es gibt keine Laufwerksbuchstaben wie zum Beispiel unter Windows. In diesem Baum kann man mit dem Kommando `cd` navigieren.

Siehe auch Linux-Fibel: Linuxfibelf - Das Dateisystem



Damit überhaupt ein hierarchischer Baum angelegt werden kann, wird ein so genanntes Basis-Filesystem (Root-Filesystem) benötigt, welches die Ablage von Daten auf einem physikalischen Datenträger regelt.

Unter Linux kommen je nach Anwendung oder Vorliebe verschiedene Dateisysteme in Frage. Kurzbeschreibung der wichtigsten Filesysteme unter Linux:

ext2fs	Das ext2 oder auch „second extended filesystem“ war viele Jahre lang das Standarddateisystem von Linux und ist immer noch weit verbreitet. Hauptnachteil von ext2 ist, dass es kein Journalingdateisystem ist.
ext3fs	Das ext3 „third extended filesystem“ ist ein Journalingdateisystem, das für das Linux entwickelt wurde. Bei vielen Linux-Distributionen ist es das Standard-Dateisystem.
reiserfs	reiserfs ist ein Mehrzweck-Dateisystem, das von einer Entwicklergruppe um Hans Reiser in der ihm gehörenden Firma Namesys unter der GPL entwickelt und realisiert wurde. ReiserFS ist das erste Journalingdateisystem, das im Linux-Kernel standardmässig (ab Kernel-Version 2.4.1) enthalten ist.
ntfs	NTFS steht für „New Technology File System“ und ist das Dateisystem von Windows NT, einschliesslich seiner Nachfolger Windows 2000, Windows XP, Windows Server 2003 und Windows Vista.
iso9660	Standard der International Organization for Standardization (ISO), der ein Dateisystem für CD-ROM-Medien beschreibt.
fat	Linux-Implementierung des FAT32-Filesystems von DOS.



Untersuchen Sie mit Hilfe des nachstehenden Textes „Filesystem Standard unter Linux“ die Filestruktur Ihres Linux-System. Starten Sie zu diesem Zweck die Maschine `vmLS1` und melden Sie sich als `vmadmin` an.

Den Befehl `ls` zum Betrachten des Verzeichnisinhaltes kennen Sie bereits. Zum Navigieren im Verzeichnisbaum benötigen Sie zusätzlich den Befehl `cd`.

Beispiel

Sie möchten von irgendwo in das Verzeichnis `/etc` wechseln:

```
vmadmin@vmLS1:~> cd /etc
```

Sie möchten in die oberste Ebene `/` des Verzeichnisbaumes wechseln:

```
vmadmin@vmLS1:~> cd /
```

Sie möchten von der aktuellen Ebene eine Stufe höher wechseln:

```
vmadmin@vmLS1:~> cd ..
```



Filesystem Standard unter Linux

Sobald der Kernel aktiv ist, lädt er als erstes das Root-Dateisystem, in dem alle für die Aufgaben des Kernels notwendigen Programme und Konfigurationsdateien angesiedelt sein müssen.

Es gibt kein klassisches Konzept für eine Datei-Hierarchie unter Linux. Seit einiger Zeit gibt es jedoch einen Filesystemhierarchiestandard (FSHS). Die meisten neueren LinuxDistributionen halten sich an diesen.

Der FSHS teilt die Daten in vier Gruppen:

- teilbare (shareable) Daten, von verschiedenen Usern/Systemen nutzbar
- unteilbare (unshareable) Daten, nur für das lokale System verwendbar
- feste (static) Daten, Programme
- variable Daten

Um diese Einteilung etwas zu verdeutlichen hier ein paar Beispiele:

- Das Home (`/home`) Verzeichnis ist teilbar (shareable), sprich ich kann in einem Netzwerk dieses Verzeichnis ohne Problem von verschiedenen Rechnern aus nutzen. Gleichzeitig ist es aber auch fest (static), da ich als normaler User z.B. keine Daten in diesem Verzeichnis ändern oder löschen kann.
- Das Lock (`/var/lock`) Verzeichnis kann hingegen immer nur für ein System verwendet werden. Es ist also unteilbar mit anderen Systemen (unshareable), da hier immer nur lokale Ressourcen blockiert (gelocked) werden z.B. das Modem.

	shareable	Unshareable
static	<code>/usr, /home</code>	<code>/etc, /boot</code>
variable	<code>/var/spool/mail,</code> <code>/var/spool/news</code>	<code>/var/run, /var/lock</code>

Das Root Verzeichnis `/`

ist das Hauptverzeichnis eines Linux Verzeichnisbaumes. Alle anderen Verzeichnisse und Dateisystemzweige sind diesem untergeordnet bzw. werden dort eingebunden (gemountet).

Was gehört zu einem Root Verzeichnis:

- alle Programme zum Starten des Systems
- alle Konfigurationsdaten um das System zu starten
- Programme zum Wiederherstellen eines defekten Bootsystems

- Backup Programme.

Das Root Verzeichnis, und die dort gespeicherten Programme, haben also lediglich die Aufgabe das Betriebssystem zu starten und bei Bedarf den "Rest" des Systems zu warten. Deshalb ist es auch relativ klein und passt bei Linux z.B. auch auf 2 Disketten. Zum Root Verzeichnis gehören also nicht nur / sondern auch die Verzeichnisse:

/bin	Programme zur Diagnose, Reparatur
/boot	boot loader Informationen
/dev	Information zur Hardware (Harddisk, CD-Rom usw.)
/etc	Information zum Starten des Systems, Konfigurationsdateien
/home	User Home Verzeichnis
/lib	Shared libraries (libc, usw.)
/media	Mount point für andere Partitionen (cdrom, disk)
/proc	Process information pseudo-filesystem
/root	Home Verzeichnis für Root
/sbin	Startprogramme (init)
/tmp	Temporary Files
/usr	zweites Hauptverzeichnis
/var	Variable Daten

Das /usr Verzeichnis (usr = unix system resources)

Im Linux Filesystem Standard findet sich folgende Beschreibung:

- /usr ist das zweite Hauptunterverzeichnis des Filesystems
- /usr ist shareable, read-only

/usr/X11R6	X Windows System, Version 11 release 6
/usr/X386	X Windows System, Version 11 release 5 on x86
/usr/bin	user commands
/usr/doc	Doku's für Programme
/usr/etc	Systemweite Konfigurationsdaten
/usr/include	Header files für den gcc
/usr/lib	Libraries
/usr/local	Lokale Hierarchie
/usr/man	online Manuals
/usr/src	Source code, z.B. der Linux Kernel

In /usr sind also alle Daten, Programme und Libraries gespeichert die man zum Arbeiten unter Linux braucht. Ebenso befinden sich hier die `man`-Pages sowie die Dokumentationen zu verschiedenen Programmen. Unter Linux ist es üblich, dass hier nur die Programme welche mit der Distribution geliefert wurden, installiert werden. Alle anderen Programme die vom Anwender nachträglich installiert werden, sollten in /usr/local/ gespeichert werden. Somit ist gewährleistet das ein Update der Distribution relativ einfach möglich ist. Das /usr/local/ Verzeichnis ist ebenso aufgebaut wie /usr, lediglich die Verzeichnisse für X11, sind hier nicht noch einmal angelegt.

Das /var Verzeichnis

/var enthält alle variablen Daten, so z.B. spool-Verzeichnisse, log-Files, usw.

/var/adm	System admin data
/var/catman	lockle formatierte <code>man</code> -Pages
/var/lib	Programm Informationen
/var/lock	Lock files
/var/log	Log files
/var/named	DNS files
/var/nis	NIS database files
/var/run	Files relevante to running processes
/var/spool	z.B. News, UUCP, Mail usw.
/var/tmp	temporäre Daten

In `/var` werden also alle variablen Daten gespeichert, da wie schon angesprochen das `/usr` Verzeichnis read-only sein sollte. Bei vielen Distributionen gibt es Links von `/usr` nach `/var` so z.B. ist `/usr/spool` ein Link auf `/var/spool`, warum sollte jetzt kein Geheimnis mehr sein.

Zusammenfassung

Der Linux Filesystem Standard bringt Ordnung in das Dateisystem und jeder der ein System administriert, sollte sich damit auseinandersetzen.



Gerät in Verzeichnisbaum einbinden

Wo befindet sich aber jetzt eine soeben eingelegte CD? Bevor auf die Daten einer CD zugegriffen werden kann, muss eine Verbindung zu einem existierenden Verzeichnis hergestellt werden. Hier kommt das Kommando `mount` ins Spiel, welches uns ermöglicht ein Gerät wie das CD-Laufwerk in den Verzeichnisbaum einzubinden (zu mounten).

Wichtig ist zu wissen, dass man zum Mounten von Geräten die Rechte dazu besitzen muss. Im produktiven Einsatzbereich ist dieser Vorgang einzig dem Administrator - also dem Benutzer `root` - vorbehalten. Es ist jedoch möglich diesen Befehl für weniger kritische Geräte wie CD-Laufwerk oder Floppy an den User zu delegieren.

Siehe auch [Linux-Fibel](#) [Linuxfibel - Das Dateisystem - Floppy, CD & Co](#)



Legen Sie eine CD in das Laufwerk und hängen Sie dieses in den Verzeichnisbaum ein. Listen Sie den Inhalt der CD mit dem Befehl `ls -al` auf. Hängen Sie das CD-Laufwerk am Schluss wieder aus.

A large grid of 20 columns and 10 rows, intended for drawing.

Beim Starten haben Sie sich als `vmadmin` angemeldet. Dieser Benutzeraccount hat jedoch nur wenig Rechte auf dem System und aus diesem Grund den Befehl sollte eigentlich der Befehl `mount` nicht ausgeführt werden können. Schauen Sie sich einmal die Datei `/etc/fstab` auf Ihrem System an und versuchen Sie mit Hilfe des Links <http://de.wikipedia.org/wiki/Fstab> herauszufinden, warum Sie als `vmadmin` den Befehl trotzdem benutzen können.

[illegible]

Dateien und Verzeichnisse manipulieren

Siehe auch [Linuxfibel - Nutzerkommandos - Datei & Verzeichnis](#)

Stellen Sie sicher, dass Sie für folgende Übung als `vmadmin` angemeldet sind und dass Sie sich in Ihrem Homeverzeichnis befinden!

```
vmadmin@vmLS1:~>
```



Führen Sie folgende Aktionen an Ihrer Konsole aus und notieren Sie die dazugehörigen Befehle:

- erstellen Sie ein Verzeichnis `linux_crash`
- wechseln Sie in das Verzeichnis
- erstellen Sie eine Datei `linux_befehle` mit folgendem Inhalt (fügen Sie einen Tabulator zwischen den Befehl und die Beschreibung):
 - `man` Hilfebefehl
 - `cat` Datei anzeigen oder neu erzeugen
 - `rm` Datei löschen
 - `cd` Verzeichnis wechseln
 - `ls` Verzeichnis anzeigen
 - `mount` Dateisystem in Verzeichnisbaum einhängen
 - `less` Datei seitenweise anzeigen
 - `mv` Datei umbenennen
- sortieren Sie den Inhalt der Datei `linux_befehle` in die Datei `linux_befehle_s`
- löschen Sie die Datei `linux_befehle`
- wechseln Sie wieder in Ihr Homeverzeichnis

A large grid of graph paper with 20 columns and 10 rows. The grid is composed of small squares, with a slightly larger square at the top left corner, likely for a title or header. The grid is empty and ready for use.

Der Name der Datei `linux_befehle_s` ist zuwenig aussagekräftig. Benennen Sie die Datei zu `linux_befehle_sort` um. Suchen Sie den entsprechenden Befehl und führen Sie die Aktion durch.

[illegible]

Erstellen Sie eine Datei mit dem Verzeichnisinhalt von /etc/Inhalt

```
vmadmin@vmLS1:~> ls -al /etc > etc inhalt
```

Verschieben Sie nun die Datei `etc.inhalt` in das Verzeichnis `~/linux` `crash`.

[illegible]



Kapiteltest 2

Bei Aufgaben, die an der Konsole zu lösen sind, notieren Sie als Lösung die korrekte Befehlsfolge.

1. Warum können Sie eine Diskette sowohl mit `mount /dev/fd0` wie auch mit `mount /media/floppy` in den Verzeichnisbaum einhängen?

[illegible]

2. Kopieren Sie die Datei `/etc/fstab` in das Verzeichnis `~/linux_crash` und benennen Sie die Kopie mit `fstab_k`. Betrachten Sie den Inhalt der kopierten Datei mit dem Befehl `cat`. Achten Sie darauf, dass der Befehlsaufbau unabhängig Ihrer momentanen Position im Filesystem ist.

[illegible]

3. Erstellen Sie ein Verzeichnis `~/Archiv` und kopieren Sie den gesamten Inhalt von `~/linux_crash` in das neue Verzeichnis.

[illegible]

4. Sie erhalten den Auftrag ein neues Programm zu installieren. An welcher Stelle des Filesystems würden Sie die entsprechenden Dateien installieren.

[illegible]

Kapitel 3 – find und grep – Die Sherlock Holmes



Lernziele

Nach diesem Kapitel sind Sie in der Lage Objekte im Filesystem zu finden:

- Sie können das `find`-Kommando nutzbringend einsetzen
- Sie verstehen die Funktionsweise des `grep`-Kommandos
- Sie können die wichtigsten regulären Ausdrücke einsetzen



Übersicht

Je nach Linux-Distribution können sich bis zu 500'000 Dateien und Verzeichnisse auf Ihrem System befinden. Um sich in diesem Dschungel zurechtzufinden benötigen wir einige Hilfsmittel die uns beim Auffinden und Erforschen dieser Objekte unterstützen können.



find

Mit dem `find`-Kommando können Sie beliebige Dateien und Verzeichnisse im Filesystem auffinden. Der Befehl ist ein wenig kompliziert, so dass Sie sich eine Zeit lang eingewöhnen müssen. Dafür ist der Befehl aber sehr mächtig und kann Ihnen gute Dienste leisten. Der Befehl hat folgenden Aufbau:

```
find <Startverzeichnis> <Suchoptionen> [-Option]
```

Die wichtigsten Suchoptionen wollen wir anhand von Beispielen untersuchen.

Beispiel

Auf welche Dateien im Verzeichnis `~/` haben Sie in der letzten Lektion (vor sieben Tagen) zugegriffen?

```
vmadmin@vmLS1:~> find ~/ -atime 7
```

Beispiel

Auf welche Dateien im Verzeichnis `~/` haben Sie von gestern bis heute zugegriffen?

```
vmadmin@vmLS1:~> find ~/ -atime -1
```

Beispiel

Auf welche Dateien im Verzeichnis `~/` haben Sie bis vor einer Woche zugegriffen?

```
vmadmin@vmLS1:~> find ~/ -atime +7
```

`-atime [+ -] i` Nach einem bestimmten Zugriffsdatum in **i** (**i** · 24 h) Tagen suchen. Wird das Pluszeichen benutzt, wird nach Dateien gesucht, auf die vor mehr als **i** · 24 h zugegriffen wurden. Bei einem Minuszeichen werden die Dateien gesucht, auf die vor weniger als **i** · 24 h zugegriffen wurde. Wird kein Plus- oder Minuszeichen angegeben, werden nur die Dateien gesucht, auf die vor genau **i** · 24 h Tagen zugegriffen wurde.

Beispiel

Auf welche Dateien im Verzeichnis `~/` haben Sie in den letzten 45 Minuten zugegriffen?

```
vmadmin@vmLS1:~> find ~/ -amin -45
```


`-amin [+ -] i` Gleich wie `-atime`, nur dass **i** als Minute gerechnet wird.



Die Suchoptionen `-mtime` und `-mmin` ermöglichen das Auffinden von geänderten Dateien. Das Suchverhalten ist sonst gleich wie bei `-atime` und `-amin`.



Die nächsten Beispiele benötigen mehr Rechte als Sie als `vmadmin` haben. Zu diesem Zweck können Sie sich temporär mit dem Befehl `su` als `root` anmelden. Achtung! Die Kennworteingabe wird nicht angezeigt.

```
vmadmin@vmLS1:~> su   
Password: gibbiX12345  
vmLS1:/home/vmadmin #
```

Das Zeichen `#` am Ende des Prompts zeigt Ihnen an, dass Sie im Moment als `root` angemeldet sind.

Beispiel

Sie suchen die Datei `nsswitch.conf`, wissen jedoch nicht wo diese sich befindet.

```
vmLS1:/home/vmadmin # find / -name nsswitch.conf
```

`-name muster`

Nach einem bestimmten Dateinamen suchen. Es können auch so genannte Jokerzeichen benutzt werden, wenn nicht der ganze Name bekannt ist.

- `*` für ein, kein oder beliebig viele Zeichen
- `?` für genau ein Zeichen

Unter Linux geht noch viel mehr. Das Zauberwort heisst „reguläre Ausdrücke“. Über die *Regular expression* gibt es ganze Bücher. Wir können uns hier nur mit den einfachsten Ausdrücken auseinandersetzen.

- `[abc]` eines der angegebenen Zeichen
- `[a-g]`, `[1-5]` ein Zeichen aus dem angegebenen Zeichen- oder Zahlenbereich.
- `[!abc]` keines des angegebenen Zeichen
- `[a, b, e-g, 3-7, 9, 0]` Kombination der Filter
- `~` steht für das Homeverzeichnis eines Benutzers

Beispiel

Sie suchen im Verzeichnis `/etc` alle Dateien die mit `n` beginnen und mit `conf` aufhören.

```
vmLS1:/home/vmadmin # find /etc -name [n]*conf
```

Siehe auch

Linuxfibel - Nutzerkommandos - Datei & Verzeichnis - find



Grep

Mit dem `grep`-Kommando können Sie innerhalb von Textdateien nach Zeichenketten suchen.

```
grep <Optionen> <Zeichenfolge> <Datei>
```

Findet `grep` das Gesuchte, werden die entsprechenden Zeilen ausgegeben. Zeichenfolgen können auch als reguläre Ausdrücke angegeben werden. Das macht den Befehl sehr mächtig.

Lassen Sie einmal mit dem Befehl `less` das Logfile `/var/log/messages` anzeigen. Jetzt suchen Sie alle Zeilen wo etwas mit `eth` vorkommt. Wahrscheinlich werden Sie einige Einträge finden. Wie wissen Sie jedoch genau ob noch andere Einträge vorhanden wären?

Beispiel

Versuchen wir das vorangehende Beispiel mit dem Befehl `grep` lösen.

```
vmLS1:/home/vmadmin # grep eth* /var/log/messages
```


Jetzt werden alle Zeilen ausgegeben welche den Ausdruck `eth0` enthalten. Wie steht es mit der Unterscheidung von Gross- und Kleinschreibung? Versuchen wir es einmal mit der Option `-i`.

```
vmLS1:/home/vmadmin # grep -i ETH0 /var/log/messages
```

Mit dieser Option werden nun auch die Zeilen mit `eth0` ausgegeben. Jetzt möchten wir jedoch nur die Zeilen vom heutigen Tag. Alle Zeilen im Log beginnen mit dem Monat und dem Tag (z.B. Aug 29). Wie können wir aber zwei Sachen kombinieren? Wir übergeben einfach das Resultat des ersten `grep` mit einer Pipe an ein zweites `grep`!

```
..vmadmin # grep -i eth0 /var/log/messages | grep "Aug 30"
```

Es scheint zu klappen, oder?



Der Befehl `grep` wird übrigens oft auch im Zusammenhang mit anderen Befehlen benutzt. So können Sie die Ausgabe eines Befehls nach einem bestimmten Suchstring durchsuchen lassen.

Zeigen Sie mit dem Befehl `ps -ax` alle laufenden Prozesse an. Was aber, wenn Sie eigentlich nur wissen möchten was auf der Konsole 1 läuft?

Beispiel

Kombinieren wir den Befehl `ps -ax` mit `grep`:

```
vmLS1:/home/vmadmin # ps -ax | grep -i tty1
```


Wir erhalten nur noch diejenigen Zeilen in welchen das Wort `tty1` vorkommt und wir können schnell sehen, welche Prozesse auf der Konsole 1 laufen.

Siehe auch Linuxfibel - Unix-Werkzeuge - grep

Apropos Konsole 1! Sie können tatsächlich mehrere Konsolen benutzen. Weiteres können Sie der Linux-Fibel im Kapitel 4.3 nachlesen. Probieren Sie es aus!



Verlassen Sie den temporären root-Modus und wechseln Sie wieder in den normalen Benutzermodus zurück.

```
vmLS1:/home/vmadmin # exit   
exit  
vmadmin@vmLS1:~>
```

Das Zeichen `>` am Ende des Prompts zeigt Ihnen an, dass Sie sich wieder im Benutzermodus befinden.



Kapiteltest 3

Bei Aufgaben, die an der Konsole zu lösen sind, notieren Sie als Lösung die korrekte Befehlsfolge.

1. Welche Dateien in Ihrem Homeverzeichnis haben Sie während der letzten 2 Stunden verändert.

[illegible]

2. Finden Sie alle Verzeichnisse oder Dateien in Ihrem Homeverzeichnis welche mit einem grossen P beginnen.

[illegible]

3. Wechseln Sie in die Konsole 2. Melden Sie sich als `root` an. Filtern Sie alle Zeilen aus der Datei `/var/log/messages` mit dem Wort `kernel` und zählen Sie die Zeilen mit dem heutigen Datum am Anfang der Zeile (z.B. Aug 30).

[illegible]

4. Bleiben Sie in der Konsole 2. Listen Sie mit dem Befehl `ps -ax` alle laufenden Prozesse auf und filtern die Ausgabe auf Prozesse, welche auf der Konsole 2 laufen (`ttty2`).

[illegible]

5. Bleiben Sie in der Konsole 2. Finden Sie alle Dateien im Verzeichnis `/etc` welche mit `conf` enden. Übergeben Sie die Ausgabe dem Befehl `sort`. Der Befehl `sort` soll seine Ausgabe in die Datei `conf_list` in das Verzeichnis `/home/vmadmin/linux_crash` schreiben. Wechseln Sie auf die Konsole 1 und schauen Sie sich die Datei mit dem Befehl `less` an.

[illegible]

Kapitel 4 – Einpacken, auspacken, wegpacken mit tar



Lernziele

In diesem Kapitel lernen Sie die Verpackung und Komprimierung unter Linux kennen:

- Sie können `tar`-Archive erstellen
- Sie sind in der Lage `tar`-Archive zu komprimieren
- Sie wissen wie man `tar`-Archive betrachtet
- Sie können `tar`-Archive entpacken



Übersicht

Werkzeuge zur Archivierung ermöglichen das „Verpacken“ mehrerer Dateien in eine einzige Datei. Zudem besteht die Möglichkeit, die Daten zu komprimieren. Dies verkürzt die Datenübertragung und hilft Diskplatz zu sparen. Zudem werden die meisten Downloads für Linux in diesem Format angeboten.



`tar` ist neben distributionsabhängigen Packprogrammen wie `rpm` oder `apt` der Standardpacker unter Linux.

`tar` ist der Name eines im Unix-Umfeld sehr geläufigen Archivierungsprogramms. Ausserdem wird so auch das Dateiformat bezeichnet, das von diesem Programm verwendet wird. Der Name wurde aus „**t**ape **a**rchiver“ (Bandarchivierer) gebildet, da mit dem Programm ursprünglich Daten auf Bandlaufwerken gesichert wurden. Gleichzeitig ist `tar` auch das englische Wort für Teer (mit dem Programm werden Dateien unkomprimiert zu einer Datei zusammengeklebt).

`tar` bietet die Möglichkeit, Dateien sequenziell in eine einzige Datei zu schreiben, bzw. Dateien aus selbiger wieder herzustellen. Die entstehende Datei trägt die Endung `.tar` (unkomprimiert) und wird auch als Tarball (dt. Teerklumpen oder Teerkugel) bezeichnet.

Heute werden `tar`-Dateien meist komprimiert, um ihre Grösse zu reduzieren. Dazu kommen üblicherweise Unix-typische Datenkompressionsprogramme wie `compress`, `gzip` oder `bzip2` zum Einsatz. Der Ansatz, erst alle Dateien unkomprimiert aneinanderzuhängen, um sie dann zu komprimieren, wird als „solid archiving“ mittlerweile auch bei anderen Archivformaten wie etwa `RAR` genutzt. Ein komprimierter Tarball trägt üblicherweise die Endung `.tar.gz`, `.tgz`, `tar.Z`, `.tar.bz2` oder `.tbz`, je nachdem mit welchem der zuvor genannten Programme er komprimiert wurde.

Schauen wir uns einmal den Befehl `tar` etwas genauer an. Grundsätzlich ist der Befehl folgendermassen aufgebaut:

```
tar [Optionen] <Archivname>.tar <Verzeichnis/Datei>
```

Bei den Optionen wird unterschieden zwischen solchen die eine Aktion auslösen und den eigentlichen Optionen.

Aktionen

- | | |
|-----------------|---|
| <code>-c</code> | (create) erstellt ein neues Archiv |
| <code>-t</code> | (table) zeigt das Inhaltsverzeichnis des Archivs an |
| <code>-r</code> | (recreate) fügt eine Datei einem bestehenden Archiv hinzu |
| <code>-d</code> | (diff) vergleicht Dateien und stellt eventuelle Unterschiede fest |
| <code>-x</code> | (extract) extrahiert die Dateien aus dem angegebenen Archiv und kopiert sie in das aktuelle Verzeichnis, ohne die Dateien aus dem Archiv zu löschen |

Eigentliche Optionen

<code>-f <archiv-datei.tar></code>	Verwendet die angegebene Datei als Archiv (wird eigentlich immer angegeben)
<code>-v</code>	(verbose) Zeigt alle Aktionen (während des Archivierens an, mit <code>--v</code> (very verbose) kann noch mehr angezeigt werden)
<code>-z</code>	Komprimiert das angegebene Verzeichnis mit dem Packprogramm <code>gzip</code>

Beispiel

Sie möchten das Verzeichnis `~/linux_crash` packen und im Verzeichnis `~/Archiv` ablegen:

```
vmadmin@vmLS1:~> tar -cvf ~/Archiv/linux_crash.tar ~/linux_crash
```

Schauen Sie das Ergebnis mit `ls -l ~/Archiv` an.

Beispiel

Mit dem vorangehenden Beispiel haben wir die Dateien aus dem Verzeichnis `~/Archiv` nur gepackt jedoch nicht komprimiert. Wiederholen wir das ganze indem wir zusätzlich komprimieren. Wenn Sie sich schon im Homeverzeichnis von `vmadmin` befinden, können Sie die Zeichen `~/` vor dem Quell- und Zielverzeichnis weglassen.

```
vmadmin@vmLS1:~> tar -cvzf Archiv/linux_crash.tar.gz linux_crash
```

Schauen Sie das Ergebnis wieder mit `ls -l ~/Archiv` an. Vergleichen Sie einmal die Dateigrösse von `linux_crash.tar` und `linux_crash.tar.gz`!

Beispiel

Sie haben jetzt tar-Archive erstellt und auch komprimiert. Im nächsten Beispiel wollen wir nun den Inhalt der Archive anschauen.

gepacktes Archiv in Kurzform anschauen

```
vmadmin@vmLS1:~> tar -tf ~/Archiv/linux_crash.tar
```

gepacktes Archiv in Langform anschauen

```
vmadmin@vmLS1:~> tar -tvf ~/Archiv/linux_crash.tar
```

gepacktes und komprimiertes Archiv in Langform anschauen

```
vmadmin@vmLS1:~> tar -tzvf ~/Archiv/linux_crash.tar.gz
```

Beispiel

Als nächstes wollen wir unsere Pakete wieder auspacken. Hier müssen Sie beachten, ob Sie die Ursprungsdateien ersetzen wollen, oder ob Sie eine Kopie der ausgepackten Dateien in einem anderen Verzeichnis erstellen wollen.

Wir entpacken unsere `tar`-Archive in das Verzeichnis `~/Restore`. Erstellen Sie als erstes das neue Verzeichnis und wechseln anschliessend in das neue Verzeichnis.

```
vmadmin@vmLS1:~> mkdir Restore
vmadmin@vmLS1:~> cd Restore
vmadmin@vmLS1:~/Restore>
```

Mit dem folgenden Befehl entpacken wir die komprimierte Version unserer Übung.

```
vmadmin@vmLS1:~/Restore> tar -xzvf ~/Archiv/linux_crash.tar.gz
```

Im Verzeichnis `~/Restore` befindet sich nun eine exakte Kopie der vorgängig gepackten Verzeichnisse und Dateien. Schauen Sie mit `ls -l` nach.



Kapiteltest 4

Bei Aufgaben, die an der Konsole zu lösen sind, notieren Sie als Lösung die korrekte Befehlsfolge.

1. Sichern Sie das Verzeichnis `/etc` als gepackte Datei `etc.tar` in das Verzeichnis `/home/vmadmin/Archiv` (für diese Aktion müssen Sie sich mit `su` in den `root`-Modus begeben).

[illegible]

- Verlassen Sie mit `exit` den root-Modus (wichtig!). Finden Sie heraus wie gross die Datei `etc.tar` geworden ist.

[illegible]

3. Komprimieren Sie die Datei `etc.tar` im gleichen Verzeichnis nach `etc.tar.gz`. Wie gross ist nun die komprimierte Version.

[illegible]

4. Schauen Sie die Datei `etc.tar.gz` mit dem `tar`-Befehl an. Wie interpretieren Sie das Resultat?

[illegible]

5. Löschen Sie mit einem Befehl alle Dateien und Unterverzeichnisse im Verzeichnis ~/Restore

[illegible]

Kapitel 5 – Informationen zu meinem System



Lernziele

Nach diesem Kapitel sind Sie in der Lage wichtige Systeminformationen abzufragen und zu interpretieren.

- Sie können ermitteln wie viel Platz welche Dateien benötigen
- Sie wissen wie der Stand der Festplattenausnutzung abzufragen ist
- Sie sind in der Lage den Systemstatus zu überprüfen
- Sie können den verfügbaren Arbeitsspeicher anzeigen
- Sie kennen die Befehle für die Prozessüberwachung



Übersicht

Unter Windows kennen Sie sicher verschiedene Tools mit denen der Systemstatus angezeigt und überwacht werden kann. Diese Möglichkeit haben Sie auch unter Linux.



du – disk usage

Mit `du` können Sie sich die Festplattenbelegung auf der Verzeichnisebene anzeigen. Es ist sowohl möglich, die Belegung summarisch über das ganze Filesystem anzuzeigen als auch Detailangaben über alle Verzeichnisse auszugeben.

Beispiel

Sie möchten die Verzeichnisgrößen ihres Homeverzeichnis anzeigen:

```
vmadmin@vmLS1:~> du
```

Die Liste ist ziemlich unübersichtlich und kann je nach Ausgangslage sehr lang werden. Probieren Sie mal folgende Variante:

```
vmadmin@vmLS1:~> du /
```

Dies macht wenig Sinn. Es gibt jedoch zu dem Befehl einige nützliche Optionen:

- c Summiert den Plattenverbrauch in der letzten Zeile der Ausgabe
- h (human readable) zeigt die Belegung in KB oder MB an
- s (summarize) zeigt nur die Gesamtsumme an



Ermitteln Sie die Plattenbelegung des gesamten Systems (muss über `su` gemacht werden) und lassen Sie das Resultat in KB oder MB anzeigen.

[illegible]

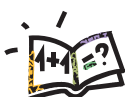
df – disk free

Der Befehl `df` macht im Grunde genommen genau das Gleiche wie der Befehl `du`, nur wird der insgesamt verbrauchte Platz eines Gerätes beziehungsweise einer Partition angezeigt.

Allerdings geht es genau um das Gegenteil: Man möchte natürlich wissen, wie viel Platz noch auf dem jeweiligen Datenträger oder der jeweiligen Partition vorhanden ist. Auch dies wird mit angezeigt. Die einzige Option ist `-h` und hat die gleiche Bedeutung wie bei `du`.

Beispiel

```
vmadmin@vmLS1:~> df -h
```



top – die Top-Ten-Liste

top zeigt eine Systemstatus-Übersicht an. Die Prozesse werden nach Systemlast an-

gegeben. Die Anzeige aktualisiert sich selbständig, solange das Programm läuft.

```

top - 21:29:46 up 12 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 55 total, 1 running, 54 sleeping, 0 stopped, 0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    250692k total,    42208k used,    208484k free,     3152k buffers
Swap:   281096k total,         0k used,    281096k free,    23316k cached

  PID USER      PR  NI    VIRT    RES    SHR  S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0   1908    848    632  S   0.0   0.3   0:01.66 init
    2 root        15  -5        0        0        0  S   0.0   0.0   0:00.01 kthreadd
    3 root        RT  -5        0        0        0  S   0.0   0.0   0:00.00 migration/0
    4 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 ksoftirqd/0
    5 root        RT  -5        0        0        0  S   0.0   0.0   0:00.00 watchdog/0
    6 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 events/0
    7 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 khelper
    8 root        RT  -5        0        0        0  S   0.0   0.0   0:00.00 kstop/0
    9 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 kintegrityd/0
   10 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 kblockd/0
   11 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 kacpid
   12 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 kacpi_notify
   13 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 cqueue
   14 root        15  -5        0        0        0  S   0.0   0.0   0:00.02 ata/0
   15 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 ata_aux
   16 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 ksuspend_usbd
   17 root        15  -5        0        0        0  S   0.0   0.0   0:00.00 khubb
   18 root        15  -5        0        0        0  S   0.0   0.0   0:00.01 kseriod

```

Ausgabe von top

Im oberen Bereich befinden sich die statistischen Gesamtangaben. Die Ausgaben sind mehr oder weniger selbsterklärend. Ein sehr wichtiger Parameter ist *load average*. Hier wird die Gesamtlast als Ganzes dargestellt. Es werden drei Werte angezeigt, wobei die beiden rechten die Durchschnittswerte der letzten Minuten sind. Der aktuelle Wert steht links. Ist er über 1.00 steht die Maschine unter Last. Ab 3.00 ist die Maschine voll ausgelastet.

Die Liste zeigt die einzelnen Prozesse nach Systemauslastung. Mit der Taste **F** erhalten Sie während der Laufzeit eine Übersicht der Optionen und können auch zusätzliche Spalten aktivieren oder deaktivieren.



Starten Sie das Programm `top`. Beobachten Sie die Ausgabe und lassen Sie sich die einzelnen Spalten in der Übersicht (Taste **F**) erklären.

In der `man`-Page können Sie sich eine ausführliche Beschreibung anzeigen lassen.

Das Programm können Sie mit der Taste **C** oder **Ctrl+C** wieder verlassen.

[illegible]

free – verfügbarer Arbeitsspeicher

Das kleine Tools `free` beschränkt seine Ausgabe auf den Arbeitsspeicher. Der Befehl kennt die zwei Optionen `-k` (Kilobyte) und `-m` (Megabyte).

```
vmadmin@umlS1:~$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	244	41	203	0	3	22
-/+ buffers/cache:		15	229			
Swap:	274	0	274			

Ausgabe von `free -m`

Lassen Sie sich nicht vom relativ kleinen freien Speicherplatz in der Spalte „free“ täuschen. Der Speicher wird dynamisch verwaltet und einige Programme greifen sich vorerst alles was verfügbar ist. Dies wird jedoch bei Bedarf wieder freigegeben, wenn es von einem anderen Prozess benötigt wird. Wenn allerdings der Swap-Speicher an seine Grenze kommt (Spalte „free“ geht gegen Null), sollte man vielleicht etwas RAM nachschieben.



ps – Die Prozessliste

Mit `ps` können die laufenden Prozesse angezeigt werden. Die Optionen können Sie sich mit `ps -help` anzeigen lassen. Der Befehl wird meist mit den Optionen `ax` oder `aux` aufgerufen.

- a alle Prozesse (nicht nur die eigenen)
- u zusätzliche Informationen wie CPU- und RAM-Benutzung
- x auch Prozesse anzeigen, die kein Terminal kontrollieren



Führen Sie den Befehl `ps aux` auf Ihrer Konsole aus und studieren Sie die Ausgabe anhand der folgenden Liste.

Überblick über die Spalten

PID	Benutzer, dem der Prozess zugeordnet ist
%CPU	Prozessorzeit, vergleiche <code>top</code>
%MEM	Speicherbelegung, vergleiche <code>top</code>
VSZ	virtuelle Prozessgrösse
RSS	Grösse des residenten Speichers (was immer im RAM bleibt)
TTY	Terminal, von wo aus der Prozess gestartet wurde
STAT	Status des Prozesses r – running, läuft aktuell s – sleeping, schläft t – terminated, gestoppt z – zombie, Vaterprozess nicht ordnungsgemäss beendet
START	Startzeit
TIME	Laufzeit
COMMAND	Befehl oder Dienst, der als Prozess läuft

[illegible]

kill – Prozesse „umbringen“

Mit `kill <PID>` kann man einen Prozess mit der angegebenen PID beenden. Es ist jedoch immer vorzuziehen, den Befehl sauber über seine eigene Beendigungsroutine zu beenden – manchmal stürzt jedoch ein Prozess ab und wir sind zum brutalen Abbruch gezwungen.

Die Brutalität des Abbruchs kann jedoch beeinflusst werden, indem dem Befehl ein bestimmtes Signal als Parameter von `kill` gesendet wird:

- 9 (SIGKILL) – Prozess wird getötet (kann Zombies hinterlassen)
- 15 (SIGTERM) – Prozess erhält die Aufforderung, sich regulär zu beenden
- HUP (SIGHUP) – Prozess wird zum Neustart aufgefordert (z.B. um eine neue Konfiguration einzulesen)

Wenn keine Option angegeben, wird das Signal SIGTERM gesendet.



Wechseln Sie auf die Konsole 2 (**Alt + F2**) und melden Sie sich als `vmadmin` an. Starten Sie das Programm `top`. Wechseln Sie dann wieder auf Konsole 1 und killen Sie den Prozess.

[illegible]



Kapiteltest 5

Bei Aufgaben, die an der Konsole zu lösen sind, notieren Sie als Lösung die korrekte Befehlsfolge.

1. Sie möchten wissen, wie gross die einzelnen Verzeichnisse und die Gesamtbelegung in Ihrem Homeverzeichnis sind (Kilo- oder Megabyte). Das Resultat speichern Sie in der Datei `home_diskbelegung` im Verzeichnis `linux_crash`. Zeigen Sie am Schluss das Resultat an.

[illegible]

2. Wie viel Platz haben Sie noch auf der Harddisk Ihrer virtuellen Maschine vLSWeb1?

[illegible]

3. Für spätere Auswertungen möchten Sie die Ausgabe von `top` während einer Minute zehn Mal in eine Datei `linux_crash/top.txt` umleiten (konsultieren Sie die `man-Page`).

[illegible]

4. Starten Sie in Konsole 2 den Befehl `less linux_crash/top.txt`. Killen Sie den Prozess von der Konsole 1.

[illegible]