

Archivieren und Extrahieren

Arbeiten im Linux-System

Agenda

- 1 Komprimierung

- 2 Archivierung

- 3 I/O-Umleitung

Komprimieren und Archivieren

Archivieren und Extrahieren

Komprimierung

- Platzbedarf für Datensatz reduzieren
 - ◆ Einsparung von Speicherplatz
 - ◆ Einsparung von über Netzwerkverbindung gesendete Daten
- Kompression = Ersetzen sich wiederholender Muster durch Daten

Angenommen in einem Roman kommen einige Wörter aus mehreren Zeichen extrem häufig vor, etwa das Wort "das". Sie könnten die Größe des Romans deutlich reduzieren, indem Sie diese mehrstelligen Wörter und Muster durch Einzelzeichen ersetzen, zum Beispiel das "das" durch einen griechischen Buchstaben, der sonst im Text nicht vorkommt.

Komprimierung

→ Zwei Varianten von Kompression: verlustfrei & verlustbehaftet

verlustfrei (lossless)	verlustbehaftet (lossy)
<ul style="list-style-type: none">→ lässt sich wieder in ursprüngliche Form überführen→	<ul style="list-style-type: none">→ können nicht vollständig wiederhergestellt werden→ z.B. für Bilder, Videos, Audiodateien (Qualitätsverlust für Menschen unmerklich, für Kontext irrelevant oder der Verlust ist den eingesparten Speicherplatz bzw. Netzwerkdurchsatz wert)

Komprimierung vs. Archivierung

→ Komprimierung

- ◆ Einsparung von Speicherplatz/Netzwerkdurchsatz
- ◆ sich wiederholende Muster werden ersetzt

→ Archivierung

- ◆ Dateien und Verzeichnisse werden in einer einzigen Datei zusammengefasst
- ◆ z.B. Backups, Source Code, Langzeitarchive

→ Archive und Komprimierung gehen meist einher: Manche Archivierungswerkzeuge komprimieren standardmäßig (andere nur optional)

- ◆ Manche Archivierungswerkzeuge müssen mit eigenständigen Komprimierungswerkzeugen kombiniert werden

Archivierung in Linux

- tar
- am häufigsten verwenden (Auf den meisten Linux-Distros eine GNU-Version von tar)
- tar archiviert, aber komprimiert nicht standardmäßig
- Linux hat auch Werkzeuge für Verwaltung von zip-Dateien, die auf Windows-Systemen üblich sind (zip/unzip).

Komprimierung in Linux

→ verlustfreie Komprimierung

- ◆ bzip2

- ◆ gzip

- ◆ Xz

- Verwenden alle unterschiedliche Algorithmen

→ Hohe Kompressionsrate → langes (De-)komprimieren

Komprimierungswerkzeuge

```
$ mkdir ~/linux_essentials-3.1
$ cd ~/linux_essentials-3.1
$ mkdir compression archiving
$ cd compression
$ cat /etc/* > bigfile 2> /dev/null

$ cp bigfile bigfile2
$ cp bigfile bigfile3
$ cp bigfile bigfile4
$ ls -lh
total 2.8M
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile2
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile3
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile4
```

Komprimierung:

```
$ bzip2 bigfile2
$ gzip bigfile3
$ xz bigfile4
$ ls -lh
total 1.2M
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile
-rw-r--r-- 1 emma emma 170K Jun 23 08:08 bigfile2.bz2
-rw-r--r-- 1 emma emma 179K Jun 23 08:08 bigfile3.gz
-rw-r--r-- 1 emma emma 144K Jun 23 08:08 bigfile4.xz
```

Dekomprimierung:

```
$ bunzip2 bigfile2.bz2
$ gunzip bigfile3.gz
$ unxz bigfile4.xz
$ ls -lh
total 2.8M
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile2
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile3
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile4
```



Wieviel Speicherplatz wird eingespart?

- ◆ Art der zu komprimierenden Datei
- ◆ Algorithmus
- ◆ Komprimierungsstufe

Komprimierungswerkzeuge

```
$ mkdir ~/linux_essentials-3.1
$ cd ~/linux_essentials-3.1
$ mkdir compression archiving
$ cd compression
$ cat /etc/* > bigfile 2> /dev/null

$ cp bigfile bigfile2
$ cp bigfile bigfile3
$ cp bigfile bigfile4
$ ls -lh
total 2.8M
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile2
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile3
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile4
```

Komprimierung:

```
$ bzip2 bigfile2
$ gzip bigfile3
$ xz bigfile4
$ ls -lh
total 1.2M
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile
-rw-r--r-- 1 emma emma 170K Jun 23 08:08 bigfile2.bz2
-rw-r--r-- 1 emma emma 179K Jun 23 08:08 bigfile3.gz
-rw-r--r-- 1 emma emma 144K Jun 23 08:08 bigfile4.xz
```

Dekomprimierung:

```
$ bunzip2 bigfile2.bz2
$ gunzip bigfile3.gz
$ unxz bigfile4.xz
$ ls -lh
total 2.8M
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile2
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile3
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile4
```

→ Wieviel Speicherplatz wird eingespart?

- ◆ Art der zu komprimierenden Datei
- ◆ Algorithmus
- ◆ Komprimierungsstufe

Komprimierungswerkzeuge

```
$ mkdir ~/linux_essentials-3.1
$ cd ~/linux_essentials-3.1
$ mkdir compression archiving
$ cd compression
$ cat /etc/* > bigfile
$ ls -lh
total 2.8M
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile2
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile3
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile4
```

Komprimierung:

```
$ bzip2 bigfile2
$ gzip bigfile3
$ xz bigfile4
$ ls -lh
total 1.2M
-rw-r--r-- 1 emma emma 712K Jun 23 08:08 bigfile
-rw-r--r-- 1 emma emma 170K Jun 23 08:08 bigfile2.bz2
-rw-r--r-- 1 emma emma 179K Jun 23 08:08 bigfile3.gz
-rw-r--r-- 1 emma emma 144K Jun 23 08:08 bigfile4.xz
```

Dekomprimierung:

```
$ bunzip2 bigfile2.bz2
$ gunzip bigfile3.gz
$ unxz bigfile4.xz
$ ls -lh
total 2.8M
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile2
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile3
-rw-r--r-- 1 emma emma 712K Jun 23 08:20 bigfile4
```

- Wenn wir mit einer Datei arbeiten wollen, müssen wir sie nicht immer direkt dekomprimieren.
- Manche Werkzeuge bieten spezielle Versionen zum z.B. Lesen von komprimierten Textdateien (gzip → cat, grep, diff, less, more, etc.)
 - ◆ dann jeweils ein z vorangestellt (bei bzip2 ein bz und bei xz das xz)
 - ◆ zcat, zgrep, ...

```
$ cp /etc/hosts ./
$ gzip hosts
$ zcat hosts.gz
127.0.0.1 localhost
```

```
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Archivierungswerkzeuge

```
$ cd ~/linux_essentials-3.1
$ tar cf archiving/3.1.tar compression
```

```
$ tar -tf 3.1.tar
compression/
compression/bigfile-xz1.xz
compression/bigfile-gz9.gz
compression/hosts.gz
compression/bigfile2
compression/bigfile
compression/bigfile-gz1.gz
compression/bigfile-xz9.xz
compression/bigfile3
```

```
$ cd ~/linux_essentials-3.1/archiving
$ ls
3.1.tar
$ tar xf 3.1.tar
$ ls
3.1.tar  compression
```

- tar ("tape archive")
- tar balls = Dateien, die mit tar erstellt worden sind
- meist wird Quellcode von Anwendungen in tar balls bereitgestellt
- Optionen:
 - ◆ -c: neue Archivdatei erstellen
 - ◆ -f: Name der zu erstellenden Datei
 - ◆ -t: Inhalt eines tar balls ansehen
 - ◆ -v: Namen der Dateien ausgeben, mit denen tar beim Erstellen/Extrahieren eines Archivs arbeitet
 - ◆ -x: entpacken einer Datei

Archivierungswerkzeuge

```
$ cd ~/linux_essentials-3.1
$ tar cf archiving/3.1.tar compression
```

```
$ tar -tf 3.1.tar
compression/
compression/bigfile-xz1.xz
compression/bigfile-gz9.gz
compression/hosts.gz
compression/bigfile2
compression/bigfile
compression/bigfile-gz1.gz
compression/bigfile-xz9.xz
compression/bigfile3
```

```
$ cd ~/linux_essentials-3.1/archiving
$ ls
3.1.tar
$ tar xf 3.1.tar
$ ls
3.1.tar  compression
```

Wir können auch nur eine Datei aus einem Archiv entpacken:

- tar ("tape archive")
- tar balls = Dateien, die mit tar erstellt worden sind
- meist wird Quellcode von Anwendungen in tar balls bereitgestellt
- Optionen:

- ◆ -c: neue Archivdatei erstellen
- ◆ -f: Name der zu erstellenden Datei
- ◆ -t: Inhalt eines tar balls ansehen
- ◆ -v: Namen der Dateien ausgeben, mit denen tar beim Erstellen/Extrahieren eines Archivs arbeitet
- ◆ -x: entpacken einer Datei

```
$ cd ~/linux_essentials-3.1/archiving
$ rm -rf compression
$ ls
3.1.tar
$ tar xvf 3.1.tar compression/hosts.gz
compression/
compression/bigfile-xz1.xz
compression/bigfile-gz9.gz
compression/hosts.gz
compression/bigfile2
compression/bigfile
compression/bigfile-gz1.gz
compression/bigfile-xz9.xz
compression/bigfile3
compression/bigfile4
$ ls
3.1.tar  compression
$ ls compression
hosts.gz
```

tar + Komprimierung

```
$ cd ~/linux_essentials-3.1/compression
$ ls
bigfile  bigfile3  bigfile-gz1.gz  bigfile-xz1.xz  hosts.gz
bigfile2  bigfile4  bigfile-gz9.gz  bigfile-xz9.xz
$ tar -czf gzip.tar.gz bigfile bigfile2 bigfile3
$ tar -cjf bzip2.tar.bz2 bigfile bigfile2 bigfile3
$ tar -cJf xz.tar.xz bigfile bigfile2 bigfile3
$ ls -l | grep tar
-rw-r--r-- 1 emma emma 450202 Jun 27 05:56 bzip2.tar.bz2
-rw-r--r-- 1 emma emma 548656 Jun 27 05:55 gzip.tar.gz
-rw-r--r-- 1 emma emma 147068 Jun 27 05:56 xz.tar.xz
```

- tar kann auch Komprimierung/Dekomprimierung von Archiven on the fly durchführen
- Verwenden die beschriebenen Komprimierungswerkzeuge
- Wir müssen lediglich eine Option beim tar-Befehl hinzufügen
 - ◆ -j: bzip2
 - ◆ -J: xz
 - ◆ z: gzip

tar + Komprimierung

```
$ cd ~/linux_essentials-3.1/compression
$ ls
bigfile  bigfile3  bigfile-gz1.gz  bigfile-xz1.xz  hosts.gz
bigfile2 bigfile4  bigfile-gz9.gz  bigfile-xz9.xz
$ tar -czf gzip.tar.gz bigfile bigfile2 bigfile3
$ tar -cjf bzip2.tar.bz2 bigfile bigfile2 bigfile3
$ tar -cJf xz.tar.xz bigfile bigfile2 bigfile3
$ ls -l | grep tar
-rw-r--r-- 1 emma emma 450202 Jun 27 05:56 bzip2.tar.bz2
-rw-r--r-- 1 emma emma 450202 Jun 27 05:56 gzip.tar.gz
-rw-r--r-- 1 emma emma 450202 Jun 27 05:56 xz.tar.xz
```

Nachträglich Dateien zu
einem (nicht komprimierten)
Archiv hinzufügen mit -u:

```
$ cd ~/linux_essentials-3.1/compression
$ ls
bigfile  bigfile3  bigfile-gz1.gz  bigfile-xz1.xz  bzip2.tar.bz2  hosts.gz
bigfile2 bigfile4  bigfile-gz9.gz  bigfile-xz9.xz  gzip.tar.gz    xz.tar.xz
$ tar cf plain.tar bigfile bigfile2 bigfile3
$ tar tf plain.tar
bigfile
bigfile2
bigfile3
$ tar uf plain.tar bigfile4
$ tar tf plain.tar
bigfile
bigfile2
bigfile3
bigfile4
$ tar uzf gzip.tar.gz bigfile4
tar: Cannot update compressed archives
Try 'tar --help' or 'tar --usage' for more information.
```

- tar kann auch Komprimierung/Dekomprimierung von Archiven on the fly durchführen
- Verwenden die beschriebenen Komprimierungswerkzeuge

Nur lediglich eine
tar-Befehl

j: bzip2
J: xz
z: gzip

Verwaltung von zip-Dateien

- Windows-Maschinen verfügen oft nicht über Anwendungen zur Verarbeitung von tar balls o.ä.
- Dort verwenden wir zip-Dateien
- zip-Datei = Archivdatei, ähnlich zu einer komprimierten tar-Datei

Verwaltung von zip-Dateien

```
$ cd ~/linux_essentials-3.1
$ mkdir zip
$ cd zip/
$ mkdir dir
$ touch dir/file1 dir/file2

$ zip -r zipfile.zip dir
  adding: dir/ (stored 0%)
  adding: dir/file1 (stored 0%)
  adding: dir/file2 (stored 0%)
$ rm -rf dir
```

```
$ ls
zipfile.zip
$ unzip zipfile.zip
Archive:  zipfile.zip
  creating: dir/
  extracting: dir/file1
  extracting: dir/file2
$ find
.
./zipfile.zip
./dir
./dir/file1
./dir/file2
```

→ zip und unzip in Linux

- ◆ Option -r sorgt dafür, dass in eine zip-Datei auch alle Unterverzeichnisse enthalten sind (Rekursion)

Befehle

- **bunzip2** Dekomprimiert eine mit bzip2 komprimierte Datei.
- **bzcat** Gibt den Inhalt einer mit bzip komprimierten Datei aus.
- **bzip2** Komprimiert Dateien mit dem bzip2-Algorithmus und -Format.
- **gunzip** Dekomprimiert eine mit gzip komprimierte Datei.
- **gzip** Komprimiert Dateien mit dem gzip-Algorithmus und -Format.
- **tar** Erstellt, aktualisiert, listet und extrahiert tar-Archive.
- **unxz** Dekomprimiert eine mit xz komprimierte Datei.
- **unzip** Dekomprimiert und extrahiert Inhalte aus einer ZIP-Datei.
- **xz** Komprimiert Dateien mit dem xz-Algorithmus und -Format.
- **zcat** Gibt den Inhalt einer mit gzip komprimierten Datei aus.
- **zip** Erzeugt und komprimiert ZIP-Archive.

Aufgabe

Dateiname	tar	gzip	bzip2	xz
archive.tar				
archive.tgz				
archive.tar.xz				

- 1. Welches der folgenden Tools wurde entsprechend der jeweiligen Erweiterung zur Erstellung dieser Dateien verwendet?**

Aufgabe

Dateiname	tar	gzip	bzip2	xz
archive.tar	X			
archive.tgz	X	X		
archive.tar.xz	X			X

- 1. Welches der folgenden Tools wurde entsprechend der jeweiligen Erweiterung zur Erstellung dieser Dateien verwendet?**

Aufgabe

Dateiname	Archiv	Komprimiert
file.tar		
file.tar.bz2		
file.zip		
file.xz		

- 1. Welche dieser Dateien sind entsprechend der jeweiligen Erweiterung Archive und welche sind komprimiert?**

Aufgabe

Dateiname	Archiv	Komprimiert
file.tar	X	
file.tar.bz2	X	X
file.zip	X	X
file.xz		X

- 1. Welche dieser Dateien sind entsprechend der jeweiligen Erweiterung Archive und welche sind komprimiert?**

Aufgabe

Wie würdest du eine Datei zu einer mit gzip komprimierten tar-Datei hinzufügen?

Aufgabe

Wie würdest du eine Datei zu einer mit gzip komprimierten tar-Datei hinzufügen?

- Dekomprimieren mit gunzip
- Neue Datei mit tar uf hinzufügen
- Komprimieren mit gzip

Aufgabe

Welche tar-Option weist tar an, den führenden / in absolute Pfade aufzunehmen?

- **-P**
- **(siehe Manpage)**

`tar -cvf backup.tar --absolute-names /home/user/docs`

`tar -cvf backup.tar -P /home/user/docs`

```
-P, --absolute-names
    Don't strip leading slashes from file names when creating archives
```

Aufgabe

Unterstützt zip verschiedene Kompressionsstufen?

- Man kann **-#** nutzen (**#** steht für eine Zahl von 0-9)
- Siehe Manpage

-#

(-0, -1, -2, -3, -4, -5, -6, -7, -8, -9)

Regulate the speed of compression using the specified digit #,

where -0 indicates no compression (store all files), -1 indicates the fastest compression speed (less compression) and -9 indicates the slowest compression speed (optimal compression, ignores the suffix list). The default compression level is -6.

Though still being worked, the intention is this setting will control compression speed for all compression methods. Currently only deflation is controlled.

Aufgabe

**Unterstützt tar beim Extrahieren
von Dateien Globs in der
Dateiliste?**

Aufgabe

Unterstützt tar beim Extrahieren von Dateien Globs in der Dateiliste?

- **Ja, mit der Option `--wildcards`. `--wildcards` muss direkt nach der tar-Datei folgen, wenn Sie die Optionen ohne Bindestrich**

verw

```
$ tar xf tarfile.tar --wildcards dir/file*  
$ tar --wildcards -xf tarfile.tar dir/file*
```

Aufgabe

Wie können wir sicherstellen, dass eine dekomprimierte Datei mit der Datei identisch ist, bevor sie komprimiert wurde?

- **Da müssen wir nicht wirklich viel tun. Alle drei bekannten Werkzeuge enthalten Prüfsummen in dem Dateiformat, die beim Dekomprimieren überprüft werden.**

Aufgabe

Was passiert wenn wir versuchen, eine Datei aus einem tar-Archiv zu extrahieren, die bereits auf unserem Dateisystem existiert?

- **Die Datei auf Ihrem Dateisystem wird mit der Version überschrieben, die sich in der tar-Datei befindet.**

Aufgabe

**Wie würden wir die Datei
archive.tgz extrahieren, ohne die
tar-Option z zu verwenden**

- **Zunächst mit gunzip
dekomprimieren**

```
$ gunzip archive.tgz  
$ tar xf archive.tar
```

Aufgabe

1. **Unterstützt tar beim Extrahieren von Dateien Globs in der Dateiliste?**
2. **Wie können wir sicherstellen, dass eine dekomprimierte Datei mit der Datei identisch ist, bevor sie komprimiert wurde?**
3. **Was passiert wenn wir versuchen, eine Datei aus einem tar-Archiv zu extrahieren, die bereits auf unserem Dateisystem existiert?**
4. **Wie würden wir die Datei archive.tgz extrahieren, ohne die tar-Option z zu verwenden**

In Dateien mit grep suchen

Archivieren und Extrahieren

Umleitung/Übergabe von Informationen

- Linux-Befehlszeile leitet Informationen über bestimmte Standardkanäle um
 - ◆ Standardeingabe (Standard Input, stdin, Kanal 0)
 - Tastatur
 - ◆ Standardausgabe (Standard Output, stdout, Kanal 1)
 - Bildschirm
 - ◆ Fehlerausgabe (Standard Error, stderr, Kanal 2)
 - Befehl o. Fehlermeldung eines Programms umleiten
- Wir können Ein- und/oder Ausgabe umleiten
- D.h. wenn wir einen Befehl ausführen, wollen wir manchmal bestimmte Informationen an den Befehl senden oder die Ausgabe in eine bestimmte Datei umleiten.

I/O-Umleitung

- I/O-Umleitung, um Informationen von oder zu einem Befehl mithilfe einer Textdatei umleiten
- Dabei können Standardeingabe, -ausgabe und -fehlerausgabe umgeleitet werden + die Informationen können aus Textdateien übernommen werden

Umleitung der Standardausgabe

```
$ echo "Hello!" > text  
$ cat text  
Hello!
```

- >
- gefolgt vom Namen der Datei. Wenn die Datei nicht existiert, wird eine neue erstellt, andernfalls werden die Informationen überschrieben

Umleitung der Standardausgabe

```
$ echo "Hello to you too!" >> text  
$ cat text  
Hello!  
Hello to you too!
```

- >>
- Um neue Informationen ans Ende der Datei hinzuzufügen
- Erstellt eine neue Datei, falls er keine existierende findet

Umleitung der Standardfehlerausgabe (Standard Error)

```
$ find /usr games 2> text-error
/usr
/usr/share
/usr/share/misc
-----Omitted output-----
/usr/lib/libmagic.so.1.0.0
/usr/lib/libdns.so.81
/usr/games
$ cat text-error
find: `games': No such file or directory
```

- 2>
- Um nur Fehlermeldungen umzuleiten
- Standardfehlerausgabe auf Kanal 2
- 2>> zum Anhängen

```
$ sort /etc 2>> text-error
$ cat text-error
sort: read failed: /etc: Is a directory
```

```
$ sort /etc/shadow 2>> text-error2
$ cat text-error2
sort: open failed: /etc/shadow: Permission denied
```

/dev/null

```
$ sort /etc 2> /dev/null
```

- Es gibt eine bestimmte Datei "Daten-Mülleimer" (bit bucket)
- Sie akzeptiert Eingaben, macht aber nichts damit
- Alle unwichtigen Informationen, die wir nicht anzeigen oder in einer Datei speichern möchten, können wir umleiten

Umleitung der Standardeingabe (Standard Input)

```
$ tr -d "l" < text
```

```
Heo!
```

```
Heo to you too!
```

Hier wird das Zeichen l
gelöscht

- <
- einem Befehl Daten aus einer bestimmten Datei statt über die Tastatur zu übergeben
- wird oft verwendet bei Befehlen, die keine Argumente akzeptieren
 - ◆ z.B. tr = dient dazu, Dateiinhalte zu übersetzen, indem man die Zeichen in einer Datei auf bestimmte Weise ändert, etwa durch das Löschen eines bestimmten Zeichens aus einer Datei

Here Documents

```
$ cat << hello
> hey
> ola
> hello
hey
ola
```

Wie im folgenden Beispiel zu sehen, dient der Operator dazu, Daten an den Befehl zu übergeben, wobei das folgende Wort keinen Dateinamen bezeichnet. Das Wort wird vielmehr als Trennzeichen der Eingabe interpretiert und nicht als Inhalt berücksichtigt; daher wird es von cat nicht angezeigt:

- <<
- Here Document = Input Stream, repräsentiert Text- oder Code-Block, der an den Befehl oder das interaktive Programm übergeben werden kann
- Skriptsprachen (bash, sh, csh) können so Eingaben direkt von der Kommandozeile übernehmen, ohne Textdateien zu nutzen

Kombinationen

```
$ find /usr admin &> newfile
$ cat newfile
/usr
/usr/share
/usr/share/misc
-----Omitted output-----
/usr/lib/libmagic.so.1.0.0
/usr/lib/libdns.so.81
/usr/games
find: `admin': No such file or directory
```

```
$ find /etc/calendar &>> newfile
$ cat newfile
/usr
/usr/share
/usr/share/misc
-----Omitted output-----
/usr/lib/libmagic.so.1.0.0
/usr/lib/libdns.so.81
/usr/games
find: `admin': No such file or directory
/etc/calendar
/etc/calendar/default
```

- &> bzw. &>> (anhängen)
- Kombination aus Kanal 1 (Standardausgabe) und Kanal 2 (Standardfehlerausgabe)

Pipes

```
$ cat /etc/passwd | less
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
:
```

Hier ändert der Befehl less nach dem Pipe-Operator die Art und Weise, wie die Datei angezeigt wird. Der Befehl less zeigt die Textdatei so an, dass der Benutzer eine Zeile auf und ab scrollen kann.

- Umleitungen meist, um Ergebnis eines Befehls zu speichern und anschließend zu verarbeiten
 - ◆ Kann ganz schön mühsam und kompliziert werden, wenn die Daten mehrere Prozesse durchlaufen.
- Um das zu vermeiden → mit Pipes | verknüpfen
- Die Ausgabe des ersten Befehls wird dann automatisch zur Eingabe des zweiten Befehls

Pipes

```
$ ls -l | head | wc -w  
10
```

- ls -l und wollen die Anzahl der Wörter aus den ersten 10 Zeilen der Ausgabe zählen
- Dazu nutzen wir head, der standardmäßig die ersten 10 Zeilen einer Datei anzeigt
- Befehl wc zählt die Wörter
 - ◆ -w: zählt nur die Wörter innerhalb der Zeilen
 - ◆ -l: würde nur die Zeilen zählen
 - ◆ -c: würde nur die Bytes zählen
- (tail würde die letzten 10 Zeilen ausgeben - ganz praktisch fürs Überwachen von syslogs, um laufende Aktivitäten zu beobachten)

- Umleitungen meist, um Ergebnis eines Befehls zu speichern und anschließend zu verarbeiten
 - ◆ Kann ganz schön mühsam und kompliziert werden, wenn die Daten mehrere Prozesse durchlaufen.
- Um das zu vermeiden → mit Pipes | verknüpfen
- Die Ausgabe des ersten Befehls wird dann automatisch zur Eingabe des zweiten Befehls

Befehle

- **cut** Entfernt Abschnitte aus jeder Zeile einer Datei.
- **cat** Zeigt Dateien an oder verknüpft sie. find Sucht nach Dateien in einer Verzeichnishierarchie.
- **less** Zeigt eine Datei an, so dass der Benutzer zeilenweise scrollen kann.
- **more** Zeigt eine Datei seitenweise an.
- **head** Zeigt die ersten 10 Zeilen einer Datei an. tail Zeigt die letzten 10 Zeilen einer Datei an.
- **sort** Sortiert Dateien. wc Zählt standardmäßig die Zeilen, Wörter oder Bytes einer Datei.

Aufgabe

- **Liste den Inhalt deines aktuellen Verzeichnisses einschließlich Eigentümer und Berechtigungen auf und leite die Ausgabe in eine Datei namens contexts.txt im Home-Verzeichnis um**

Aufgabe

```
$ ls -l > contents.txt
```

- **Liste den Inhalt deines aktuellen Verzeichnisses einschließlich Eigentümer und Berechtigungen auf und leite die Ausgabe in eine Datei namens contents.txt im Home-Verzeichnis um**

Aufgabe

- **Sortiere den Inhalt der Datei contexts.txt aus deinem aktuellen Verzeichnis und füge ihn an das Ende einer neuen Datei namens contents-sorted.txt an**

Aufgabe

```
$ sort contents.txt >> contents-sorted.txt
```

- **Sortiere den Inhalt der Datei `contexts.txt` aus deinem aktuellen Verzeichnis und füge ihn an das Ende einer neuen Datei namens `contents-sorted.txt` an**

Aufgabe

- **Zeige den Inhalt der letzten 10 Zeilen der Datei /etc/passwd an und leite sie in eine neue Datei im Verzeichnis Documents deines Benutzers um**

Aufgabe

```
$ tail /etc/passwd > Documents/newfile
```

- **Zeige den Inhalt der letzten 10 Zeilen der Datei /etc/passwd an und leite sie in eine neue Datei im Verzeichnis Documents deines Benutzers um**

Aufgabe

- **Zähle die Anzahl der Wörter in der Datei contexts.txt und hänge die Ausgabe an das Ende der Datei field2.txt in deinem Homeverzeichnis an. Du müsstest jetzt die Eingabe- als auch Ausgabeumleitung verwenden.**

Aufgabe

```
$ wc < contents.txt >> field2.txt
```

- **Zähle die Anzahl der Wörter in der Datei contexts.txt und hänge die Ausgabe an das Ende der Datei field2.txt in deinem Homeverzeichnis an. Du müsstest jetzt die Eingabe- als auch Ausgabeumleitung verwenden.**

Aufgabe

- **Zeige die ersten 5 Zeilen der Datei /etc/passwd an und sortiere die Ausgabe alphabetisch umgekehrt**

Aufgabe

```
$ head -n 5 /etc/passwd | sort -r
```

- **Zeige die ersten 5 Zeilen der Datei /etc/passwd an und sortiere die Ausgabe alphabetisch umgekehrt**

Aufgabe

- **Zähle mit der zuvor erstellten Datei contents.txt die Anzahl der Zeichen der letzten 9 Zeilen**

Aufgabe

```
$ tail -n 9 contents.txt | wc -c  
531
```

- **Zähle mit der zuvor erstellten Datei contents.txt die Anzahl der Zeichen der letzten 9 Zeilen**

Aufgabe

- **Zähle die Anzahl der Dateien namens test im Verzeichnis /usr/share und dessen Unterverzeichnissen.
Hinweis: Jede Zeilenausgabe des Befehls find steht für eine Datei**

Aufgabe

```
$ find /usr/share -name test | wc -l  
125
```

- **Zähle die Anzahl der Dateien namens test im Verzeichnis /usr/share und dessen Unterverzeichnissen.**
Hinweis: Jede Zeilenausgabe des Befehls find steht für eine Datei