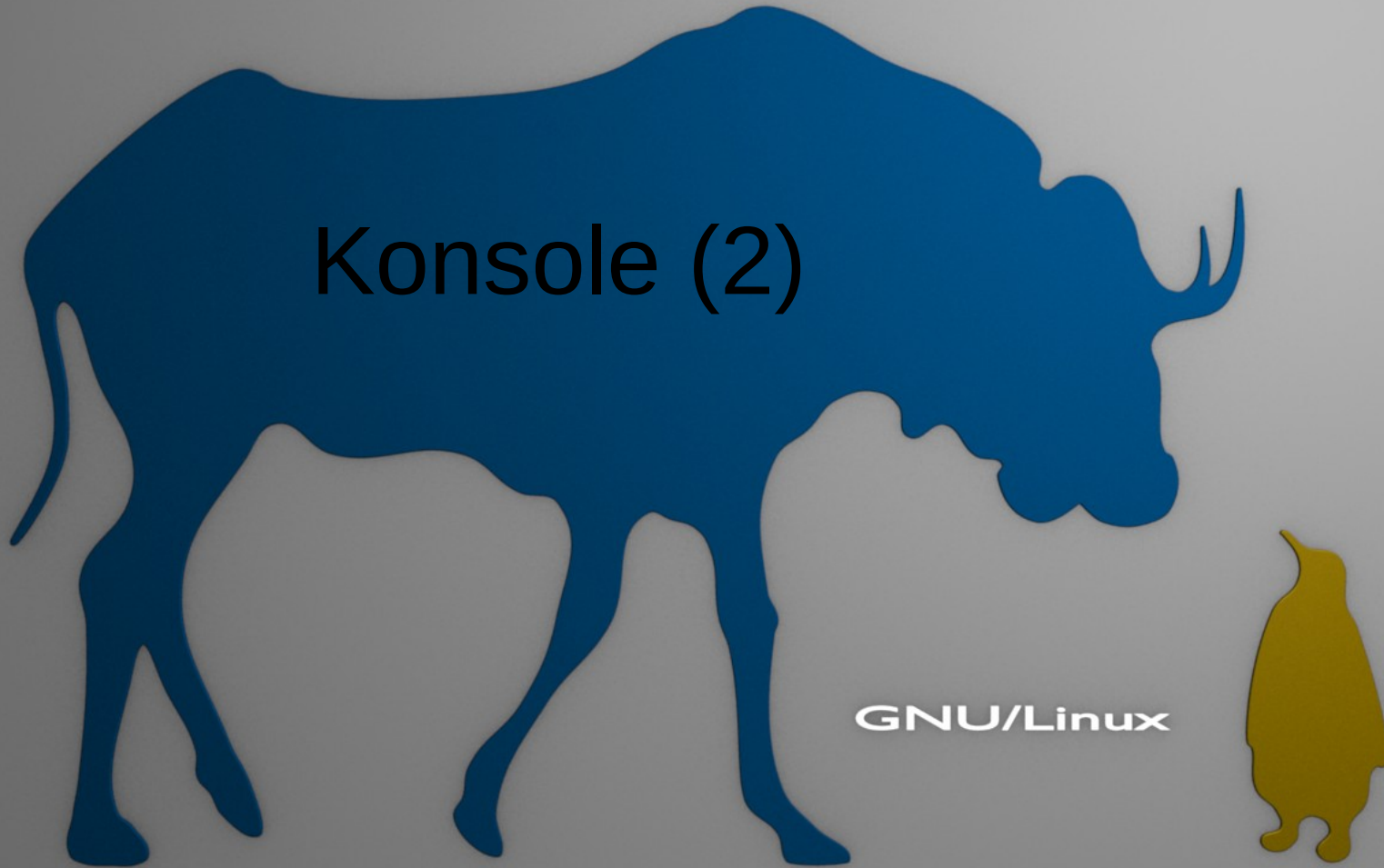


# Linux Tutorium

Konsole (2)

GNU/Linux



# Konsole

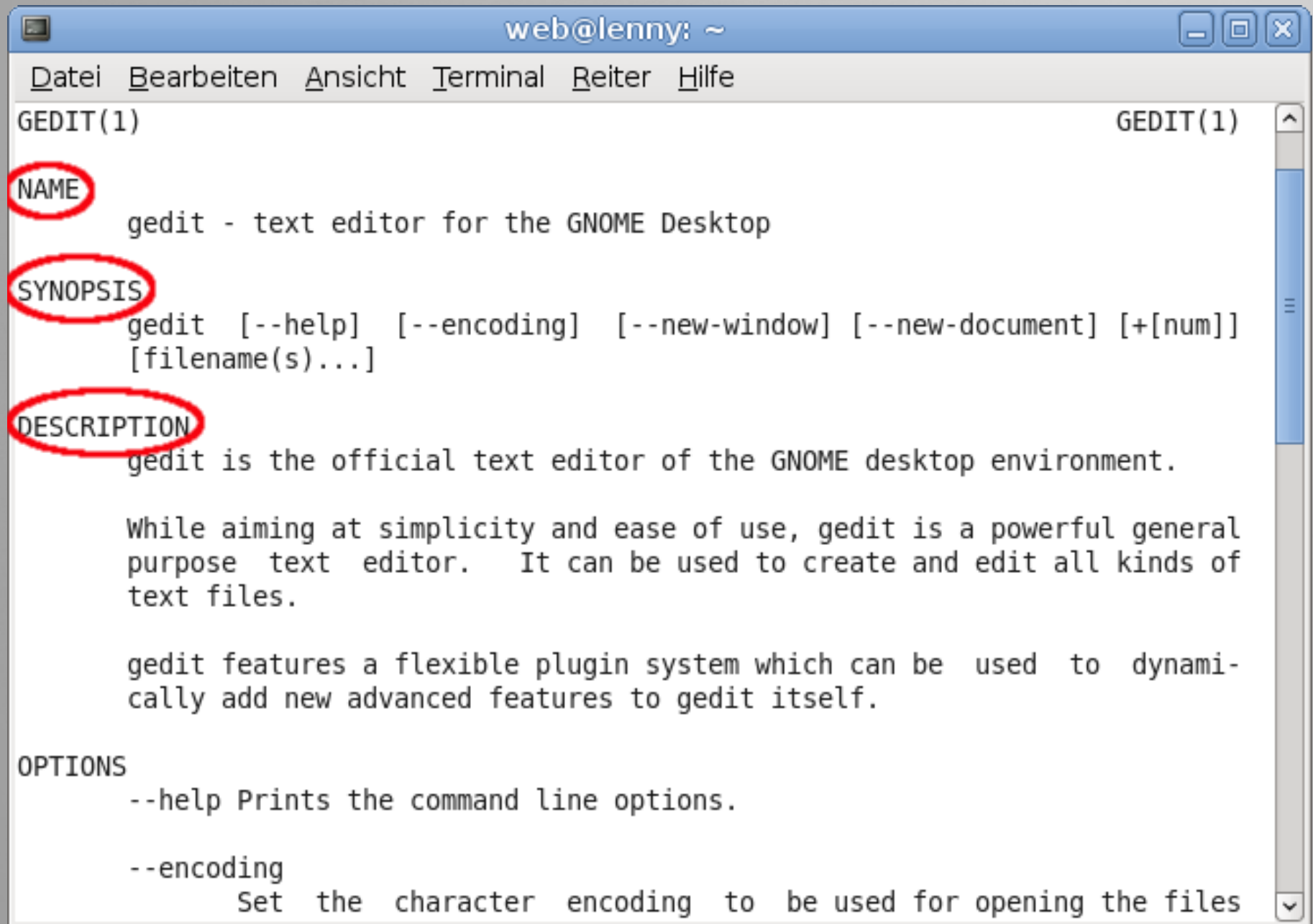
Man pages enthalten mindestens die folgenden Elemente:

- NAME
  - Name und Kurzbeschreibung
- SYNOPSIS
  - Syntaxbeschreibung
- DESCRIPTION
  - Ausführliche Beschreibung des Kommandos

GNU/Linux



# Konsole



```
web@lenny: ~
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
GEDIT(1) GEDIT(1)
NAME
  gedit - text editor for the GNOME Desktop
SYNOPSIS
  gedit [--help] [--encoding] [--new-window] [--new-document] [+num]
  [filename(s)...]
DESCRIPTION
  gedit is the official text editor of the GNOME desktop environment.

  While aiming at simplicity and ease of use, gedit is a powerful general
  purpose text editor. It can be used to create and edit all kinds of
  text files.

  gedit features a flexible plugin system which can be used to dynami-
  cally add new advanced features to gedit itself.
OPTIONS
  --help Prints the command line options.

  --encoding
        Set the character encoding to be used for opening the files
```

# Konsole

„whoami“

- Zeigt den Benutzernamen, mit dem man gerade eingeloggt ist

„id“

- Zeigt die Benutzer-ID und Gruppenzugehörigkeiten des aktuellen Benutzers

GNU/Linux



# Konsole

## Dateien und Verzeichnisse

„cd pfad“

- Wechselt in den angegebenen Pfad

„pwd“

- Zeigt das aktuelle Verzeichnis an

„clear“ bzw. Strg+l

- Löscht den Bildschirminhalt

GNU/Linux



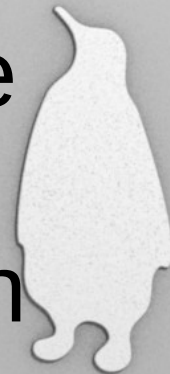
# Konsole

## Dateien und Verzeichnisse

„ls“

- Zeigt den Inhalt des aktuellen Verzeichnisses an
- Durch Parameter kann man die Ausgabe verändern
- Häufigste Parameter: l, i, s, a, h

GNU/Linux



# Konsole

## Dateien und Verzeichnisse

„fdisk -l“

- Zeigt die vorhandenen Festplatten (-partitionen) an

„df -h“

- Zeigt den freien Speicherplatz auf einem Datenträger an

GNU/Linux

# Konsole

## Dateien und Verzeichnisse

„cp quelle ziel“

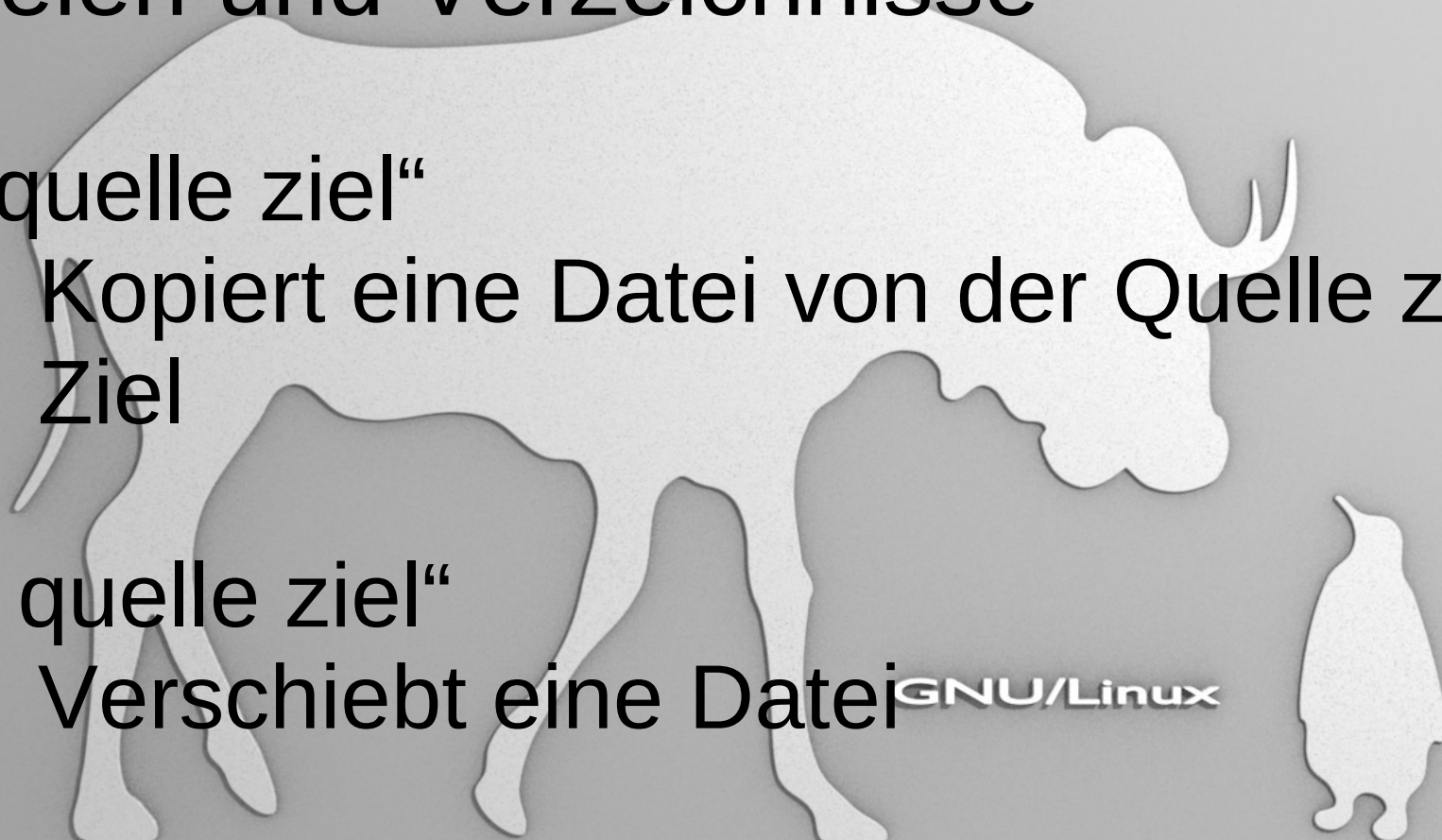
- Kopiert eine Datei von der Quelle zum Ziel

„mv quelle ziel“

- Verschiebt eine Datei

„rm datei“

- Löscht eine Datei





# Konsole

## Dateien und Verzeichnisse

Sind die Quellen Verzeichnisse, so benötigen „cp“ und „rm“ zusätzlich den Parameter „-r“

GNU/Linux



# Konsole

## Dateien und Verzeichnisse

„rmdir verzeichnis“

- Löscht ein Verzeichnis

„mkdir verzeichnis“

- Legt ein neues Verzeichnis mit dem angegebenen Namen an

GNU/Linux



# Konsole

## Dateien und Verzeichnisse

„touch datei“ bzw. „touch verzeichnis“

- Ändert den Zeitstempel der angegebenen Datei bzw. des Verzeichnisses
- Ist die Datei nicht vorhanden wird sie mit der Größe von 0 Byte angelegt
- Touch wird vorrangig zum Anlegen von Dateien verwendet

# Konsole

## Dateien und Verzeichnisse

„chmod oktalrechte datei-oder-verzeichnis“

- Ändert die Benutzerrechte

„chown benutzer[:gruppe] datei-oder-verzeichnis“

- Ändert den Besitzer

„chgrp datei-oder-verzeichnis“

- Ändert die Gruppe

GNU/Linux



# Konsole

## Suchen

„find pfad -name „begriff““

- Durchsucht den Verzeichnisbaum vom angegebenen Pfad aus nach Dateien und Verzeichnissen, die „begriff“ im Namen enthalten und zeigt diese an.

GNU/Linux

# Konsole

## Suchen

„grep -slr „begriff“ pfad“

- Durchsucht den Verzeichnisbaum vom angegebenen Pfad aus nach Dateien und Verzeichnissen, die „begriff“ als Inhalt enthalten und zeigt diese an.

GNU/Linux

# Konsole

## Arbeiten mit Archiven

### Parameter

- c - compress
- v - verbose
- z - gnu zip compression
- j - bzip2 compression
- f - following file
- x - extract

GNU/Linux



# Konsole

## Arbeiten mit Archiven

„tar -cvf archiv.tar datei1 verzeichnis/  
datei2 ...“

- Anlegen eines Tarballs

„tar -xvf archiv.tar“

- Extrahieren aus einem Tarball

GNU/Linux





# Konsole

## Arbeiten mit Archiven

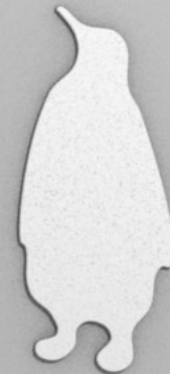
„tar -cvzf archiv.tar.gz datei1 verzeichnis/  
datei2 ...“

- Anlegen eines gzip-komprimierten Tarballs

„tar -xvzf archiv.tar.gz“

- Extrahieren aus einem gzip-komprimierten Tarball

GNU/Linux



# Konsole

## Arbeiten mit Archiven

„tar -cvjf archiv.tar.bz2 datei1 verzeichnis/  
datei2 ...“

- Anlegen eines bzip2-komprimierten Tarballs

„tar -xvjf archiv.tar.bz2“

- Extrahieren aus einem bzip2-komprimierten Tarball

GNU/Linux



# Konsole

## Ansehen und editieren von Dateien

„less datei“

- Zeigt datei in einem scrollbaren Fenster an

„cat datei“

- Zeigt den Inhalt der Datei direkt auf der Konsole aus

GNU/Linux



# Konsole

## Ansehen und editieren von Dateien

„nano datei“

- Öffnet datei im Konsoleneditor nano

„vi datei“

- Öffnet datei im Standardeditor vi

GNU/Linux

# Konsole

## Benutzermanagement

„adduser benutzername“

- Neuen Benutzer anlegen

„deluser benutzername“

- Benutzer löschen

GNU/Linux



# Konsole

## Benutzermanagement

„su benutzername“

- Nach Passworteingabe zum angegebenen Benutzer werden
- Ohne Benutzername: Wechsel zum Superuser (= root)

GNU/Linux



# Konsole

## Dateisystemmanagement

„mount“

- Dient zum Einhängen/Einbinden von Dateisystemen

„umount“

- Hängt eingebundene Dateisysteme wieder aus

GNU/Linux



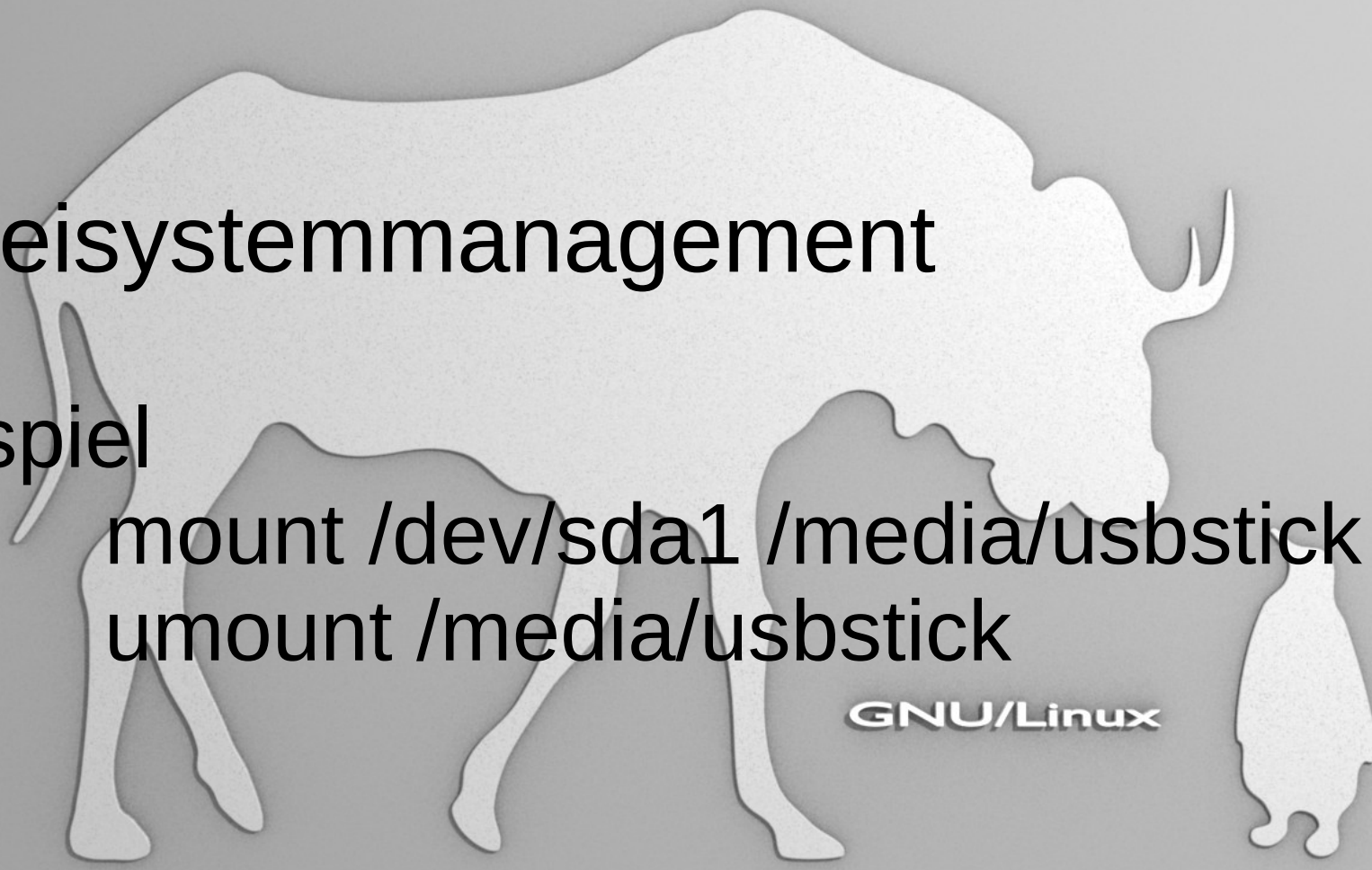
# Konsole

Dateisystemmanagement

Beispiel

```
mount /dev/sda1 /media/usbstick  
umount /media/usbstick
```

GNU/Linux





# Konsole

## I/O-Umleitungen

„< datei“

- Die Ausgabeumlenkung funktioniert durch ein Größer-als-Zeichen, wobei die Nummer des Deskriptors vorangestellt werden sollte
- Wird keine Nummer vorangestellt, wird automatisch die Standardausgabe verwendet

GNU/Linux

# Konsole

## I/O-Umleitungen

„< datei“

- Die Standardausgabe wird nach datei umgeleitet
- Die datei wird, wenn sie noch nicht vorhanden ist, neu angelegt

GNU/Linux



# Konsole

## Standarddeskriptoren

Jedem Prozess sind unter Unix standardmäßig 3 Deskriptoren zugewiesen:

- Die Standardeingabe (0, STDIN)
- Die Standardausgabe (1, STDOUT)
- Die Standardfehlerausgabe (2, STDERR)

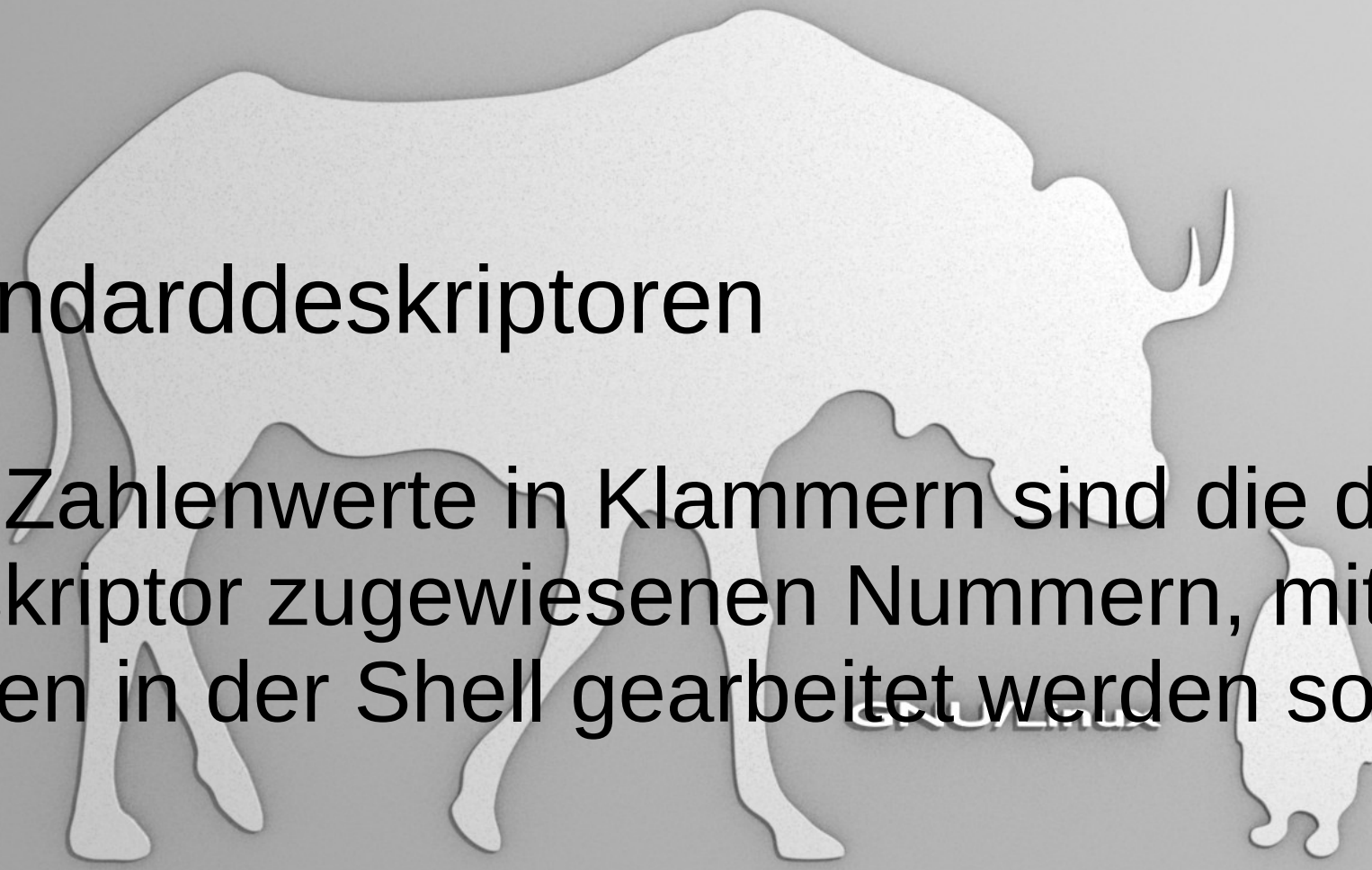
GNU/Linux

Über diese Deskriptoren kann die Ein- und Ausgabe des Programms erfolgen

# Konsole

## Standarddeskriptoren

Die Zahlenwerte in Klammern sind die dem Deskriptor zugewiesenen Nummern, mit denen in der Shell gearbeitet werden sollte



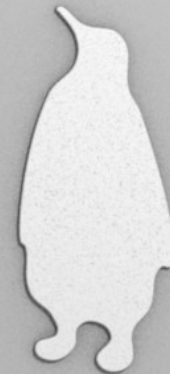
# Konsole

## I/O-Umleitungen

### Beispiele

- `cat /proc/cpuinfo > info.txt`
- `ls 1> list`

GNU/Linux



# Konsole

## I/O-Umleitungen

„<< datei“

- Manchmal kommt es vor, dass die Ausgabe in eine Datei umgelenkt werden soll, in der sich bereits ein Inhalt befindet, der nicht überschrieben werden soll
- Die Standardausgabe wird an die Datei angehängen

GNU/Linux

# Konsole

I/O-Umleitungen

Beispiel

```
uname -a >> info.txt
```

GNU/Linux



# Konsole

## I/O-Umleitungen

„> datei“

- Die Standardeingaben werden nach „datei“ umgeleitet

GNU/Linux





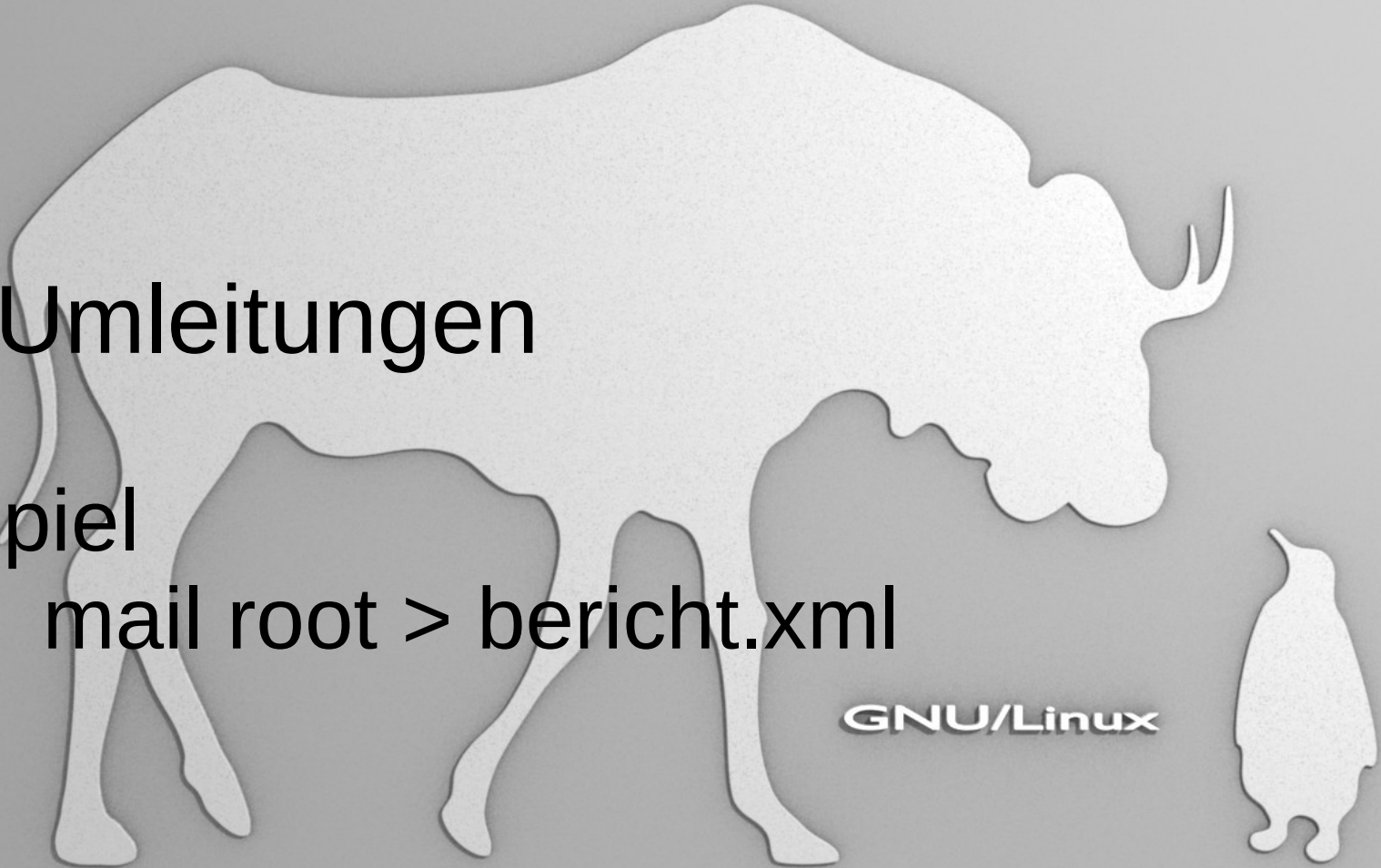
# Konsole

I/O-Umleitungen

Beispiel

```
mail root > bericht.xml
```

GNU/Linux

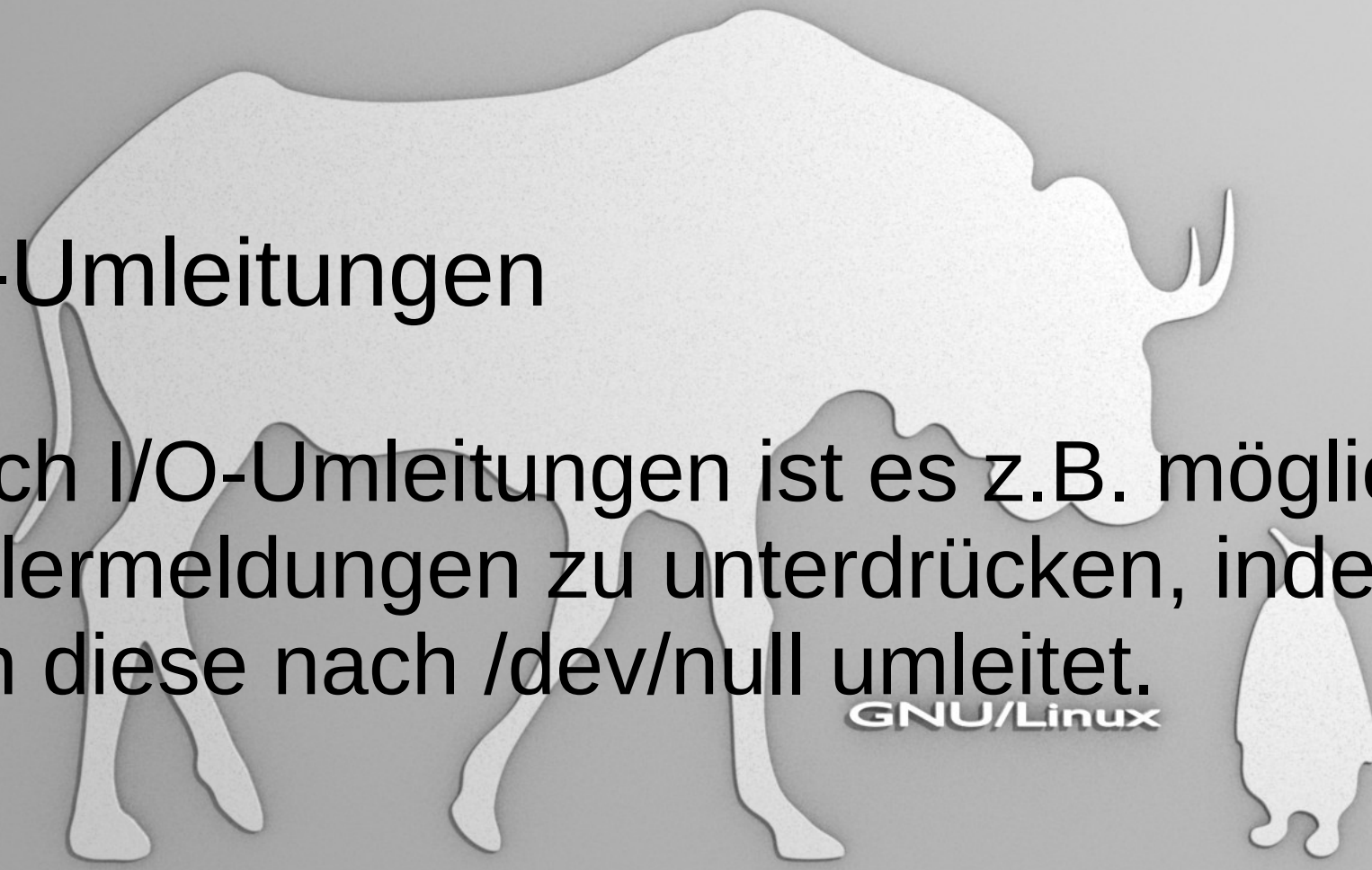


# Konsole

## I/O-Umleitungen

Durch I/O-Umleitungen ist es z.B. möglich, Fehlermeldungen zu unterdrücken, indem man diese nach `/dev/null` umleitet.

GNU/Linux



# Konsole

## Pipes

- Es ist möglich, die Standardausgabe eines Kommandos direkt einem zweiten Kommando als Standardeingabe zu übergeben
- Pipes stellen eine Art Interprozess-Kommunikation dar, da die Ausgabe eines Programms nicht in eine Datei geschrieben sondern direkt an ein Programm weitergeleitet wird

# Konsole

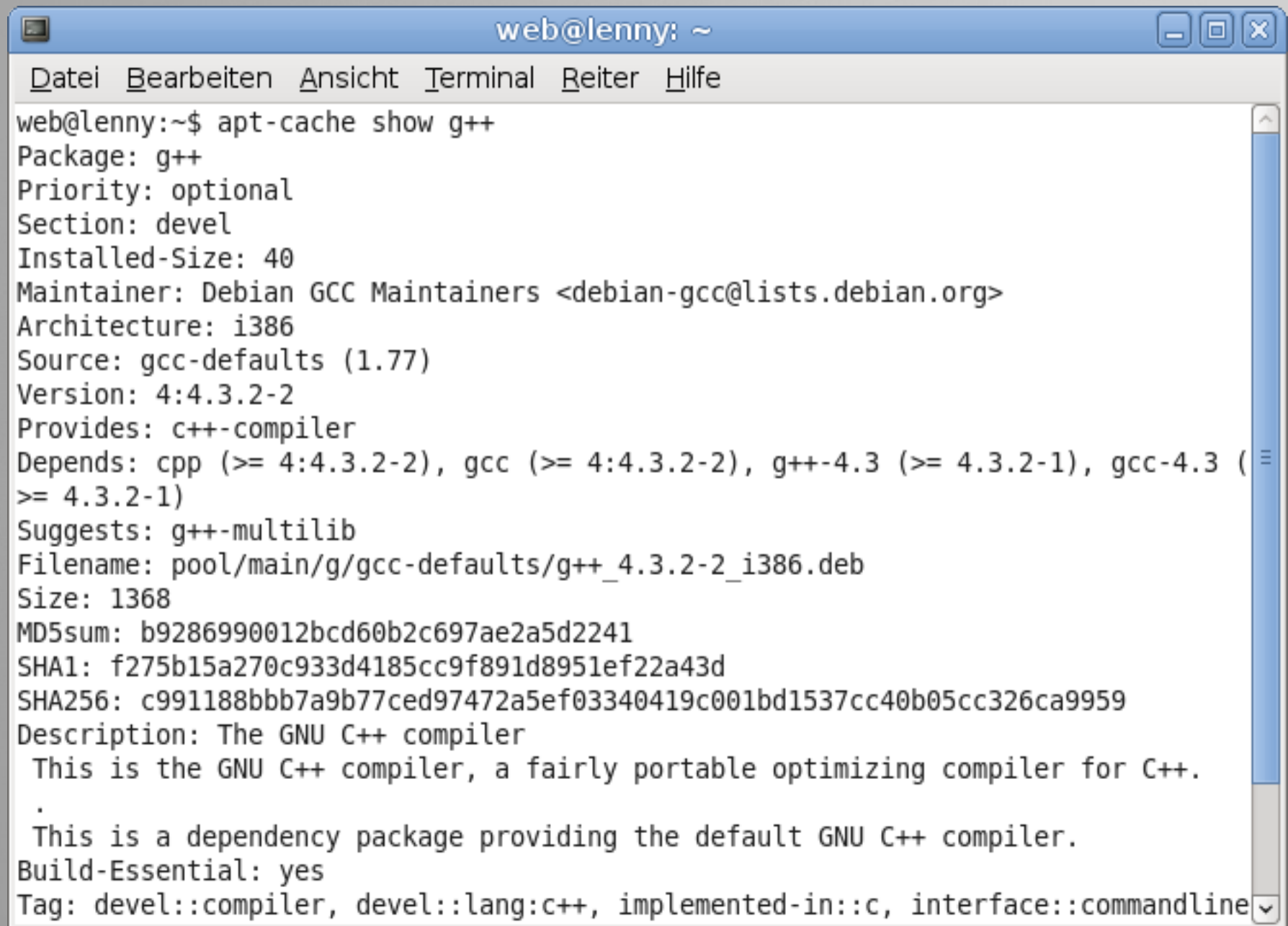
## Pipes

- Diese Weiterleitung erfolgt mit einem Pipe-Operator (|)
- Dieser Strich wird zwischen die beiden Programme gesetzt:  
Befehl1 | Befehl2

GNU/Linux

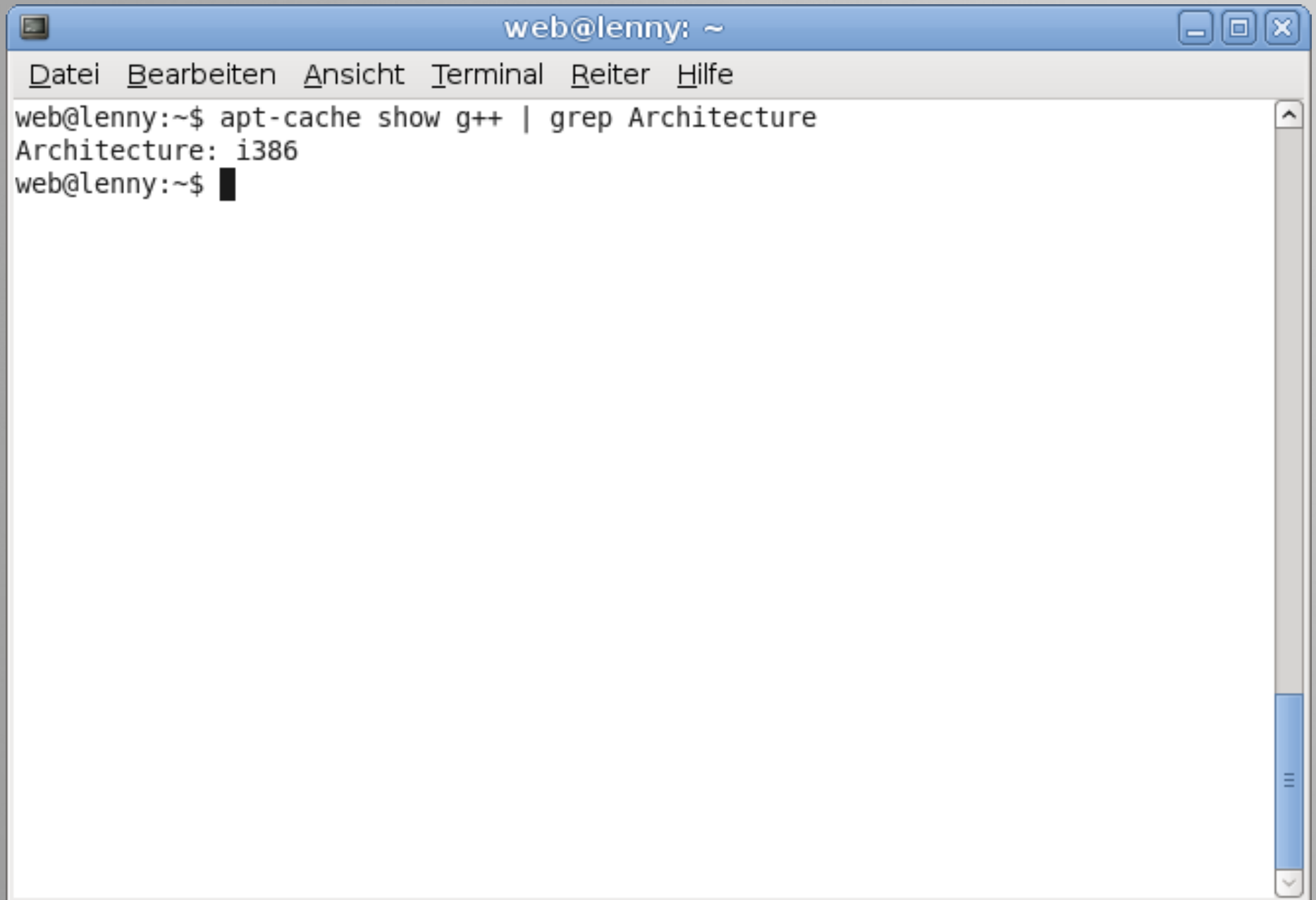


# Konsole



```
web@lenny: ~  
Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
web@lenny:~$ apt-cache show g++  
Package: g++  
Priority: optional  
Section: devel  
Installed-Size: 40  
Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>  
Architecture: i386  
Source: gcc-defaults (1.77)  
Version: 4:4.3.2-2  
Provides: c++-compiler  
Depends: cpp (>= 4:4.3.2-2), gcc (>= 4:4.3.2-2), g++-4.3 (>= 4.3.2-1), gcc-4.3 (>= 4.3.2-1)  
Suggests: g++-multilib  
Filename: pool/main/g/gcc-defaults/g++_4.3.2-2_i386.deb  
Size: 1368  
MD5sum: b9286990012bcd60b2c697ae2a5d2241  
SHA1: f275b15a270c933d4185cc9f891d8951ef22a43d  
SHA256: c991188bbb7a9b77ced97472a5ef03340419c001bd1537cc40b05cc326ca9959  
Description: The GNU C++ compiler  
  This is the GNU C++ compiler, a fairly portable optimizing compiler for C++.  
.  
  This is a dependency package providing the default GNU C++ compiler.  
Build-Essential: yes  
Tag: devel::compiler, devel::lang:c++, implemented-in::c, interface::commandline
```

# Konsole



The image shows a terminal window titled "web@lenny: ~". The window has a menu bar with "Datei", "Bearbeiten", "Ansicht", "Terminal", "Reiter", and "Hilfe". The terminal content shows the command "apt-cache show g++ | grep Architecture" being executed, resulting in the output "Architecture: i386". The prompt "web@lenny:~\$" is visible at the end of the line.

```
web@lenny:~$ apt-cache show g++ | grep Architecture
Architecture: i386
web@lenny:~$
```

# Konsole

## Bash-History

- Sie speichert die zuletzt eingegebenen Befehle
- Man kann diese also aufrufen ohne sie erneut einzugeben
- Die Bash macht die Befehle editierbar

GNU/Linux



# Konsole

## Bash-History

- Eine Übersicht über die in der History enthaltenen Befehle liefert der Befehl „history“

GNU/Linux





# Konsole

## Bash-History

- Die Cursorstasten erleichtern den Umgang mit der History
  - Durch die Cursorstaste „Nach Oben“ kann der zuletzt eingegebene Befehl erneut aufgerufen werden. So ist das Durchlaufen der History bis an den Anfang möglich
  - Invers dazu ist dies auch mit der „Nach Unten“-Taste möglich wenn man sich in der History befindet

# Konsole

## Bash-History

- Auch eine Suche in der History ist relativ einfach möglich
  - [Strg-r] Rückwärtiges Suchen im Historybuffer
  - [Strg-s] Vorwärts im Historybuffer suchen

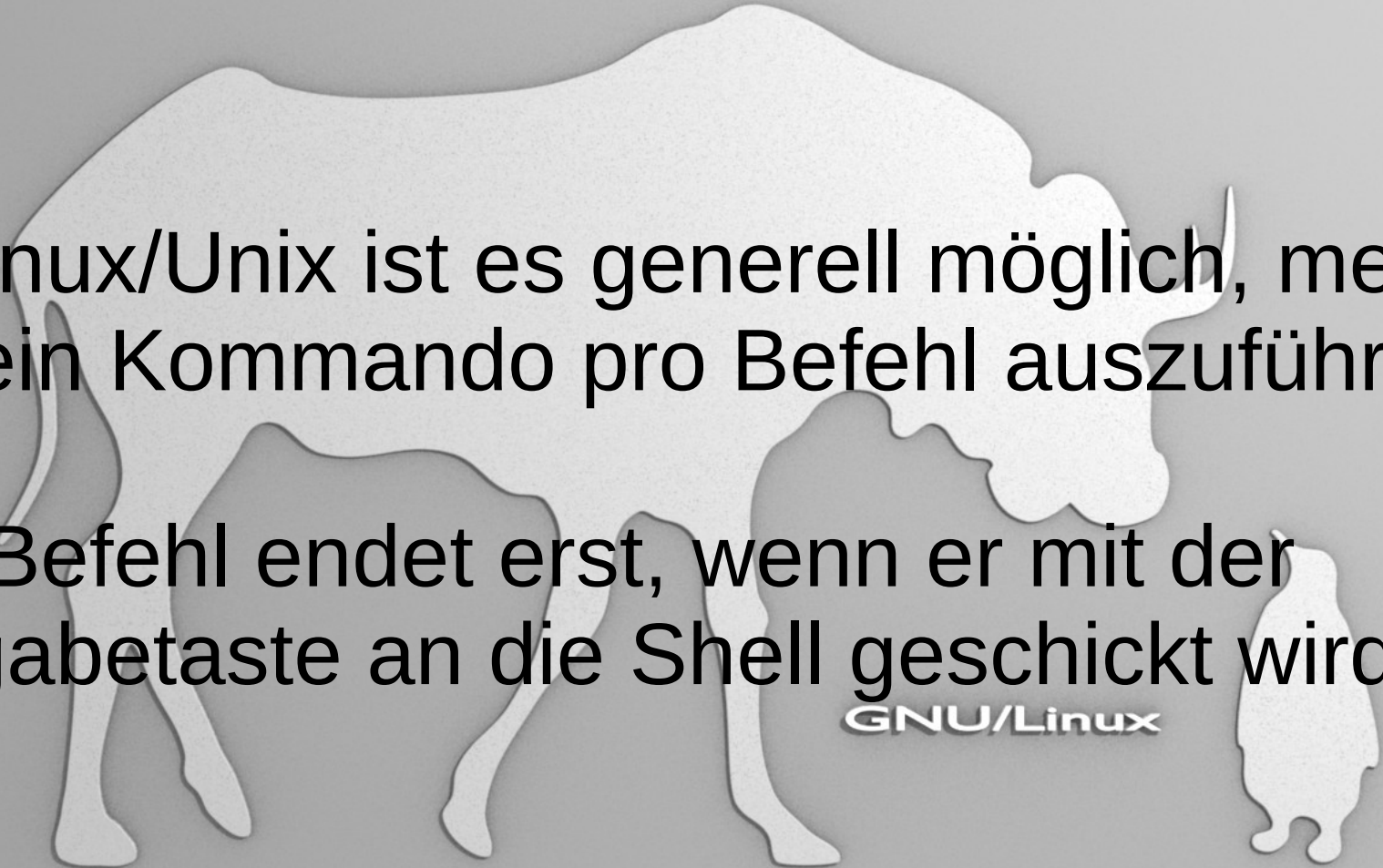
GNU/Linux

# Konsole

In Linux/Unix ist es generell möglich, mehr als ein Kommando pro Befehl auszuführen.

Ein Befehl endet erst, wenn er mit der Eingabetaste an die Shell geschickt wird

GNU/Linux



# Konsole

Ein Kommando wird von einem anderen durch ein Semikolon getrennt (;)  
Die Kommandos laufen immer hintereinander ab, auch wenn einzelne Kommandos fehlschlagen

Beispiel:

ls Verzeichnis ; uname ; find / -name datei

GNU/Linux

# Konsole

dmesg

- Anzeigen der Kontrollmeldungen des Systems aus dem Ringbuffer des Kernels
- In diesem Buffer sind alle Kernelmeldung seit dem letzten hochfahren (außer der Buffer war voll)

GNU/Linux

# Konsole

## Shell-Aliase

- Erlauben es, einen neuen Befehlsnamen zu vergeben, der stellvertretend für ein Kommando oder eine Kommandoreihe steht
- Ein Alias wird über das gleichnamige Kommando „alias“ erstellt und modifiziert

GNU/Linux

# Konsole

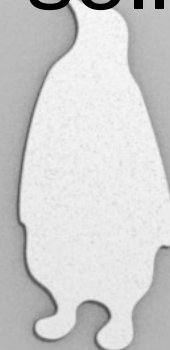
## Shell-Aliase

- Einen Alias erstellt man, indem man den neuen Befehlsnamen sowie den Befehl, der hinter diesem verborgen sein soll, dem Alias-Kommando übergibt

- Beispiel:

```
alias dir="ls -lsa"
```

GNU/Linux



# Konsole

## Shell-Aliase

- Möchte man einen Alias löschen, ist dies einfach durch die Eingabe des Befehls „unalias“ möglich, dem der zu löschende Alias übergeben wird
- Beispiel:  
unalias dir

GNU/Linux





# Konsole

## Shell-Aliase

- Eine Liste der vorhandenen Alias-Einträge kann ebenfalls durch den Befehl „alias“ (ohne weitere Parameter) ausgegeben werden
- Um den Alias dauerhaft zu speichern muss man z.B. beim Aufruf die Option „-p“ anfügen  
z.B.: `alias -p dir="ls -lsa"`

GNU/Linux