German Howto

Table of Contents

German Howto

Winfried Trümper <me@wt.xpilot.org>

v2.0.1, 29. December 2001

Dieses Dokument enthält Tips und Informationen für deutschsprachige Linux−Anwender. (Note to English readers: this document explains the german specific aspects of Linux. The middle chapter is written in English.)

---

1. Deutscher Teil (German Part)

2. English Part (Englischer Teil)

3. Fun für Dich: Denglish

---

1. Deutscher Teil (German Part)

1.1 Einleitung

Die Bezeichnung Howto ist eine Zusammenschreibung von how to, was Sie mit "so mache ich..." übersetzen können. Das Linux German Howto beschreibt also wie man deutschsprachige Eigenheiten unter Linux aktiviert. Gesprochen wird es als Dschörman Hautu. Im Folgenden schreibe ich lieber vom Deutschen Sowirdsgemacht für Linux. Schon bin ich Punk.

Was sind denn nun deutsche Eigenheiten? Ledertrachten, Weisswürste, Bier und gebügelte Unterwäsche, so wie es jedem ausländischen Besucher im Bierzelt auf der CeBIT vorgemacht wird? Hm, das wären wohl eher Anpassungen, die man an sich selbst vornimmt. Aber keine Furcht, Linux lässt sich auch, dem amerikanischen Einfluss sei dank, ganz leger in Jeans beim Pizzaessen und einem Schluck Afri−Cola bedienen (für Imis: Afri−Cola wird in Köln gebraut). Auf das Thema gebügelte Unterwäsche kann ich an dieser Stelle, dem amerikanischen Einfluss sei ebenfalls gedankt, nicht weiter eingehen. Denn obwohl es hier schon spät ist, sind dort die Kinder ja noch wach. Sagt zumindest der Sowirdsgemacht−Projektleiter. Aber genug gescherzt, sonst entfernen wir uns zu weit vom Vorurteil der unfreundlichen Deutschen.

Linux wird von einer weltweiten Gemeinde von Hackern im Internet entwickelt, deren kleinster gemeinsamer Nenner für die Verständigung Englisch ist. Dagegen müssen Anwender üblicherweise in ihrer Muttersprache und mit nationalen Eigenheiten arbeiten: zum Beispiel Zeichen, Zeitzonen oder Zahlenformaten. Erfreulicherweise hat Linux eine so weite Verbreitung erfahren, dass die nationalen Märkte längst Ziele für die Linux−Distributionen geworden sind. D.h. die regionalen Anpassungen der Anwendungen und des Systems stecken in Linux weitgehend drin. In diesem Zusammenhang möchte man das KDE−Projekt erwähnen, welches auf dem Gebiet der Internationalisierung hinsichtlich Anzahl der Sprachen und Vollständigkeit der Anpassungen hohe Maßstäbe gesetzt hat. Dieser Entwicklung trägt auch das Deutsche Sowirdsgemacht Rechnung, indem der Schwerpunkt ab sofort mehr auf der Beschreibung von Grundlagen liegt.

Urheberrechte, Lizenz und Haftungsausschluss

Text, Abbildungen und Programme in diesem Dokument wurden mit grosser Sorgfalt erarbeitet. Trotzdem sind Fehler nicht ganz auszuschliessen. Angesichts der Komplexität und der Schnelllebigkeit heutiger Computer−Systeme kann der Autor weder die Richtigkeit noch die Anwendbarkeit der gelieferten Informationen garantieren. Kurz: Benutzung auf eigene Gefahr.

Verfügbarkeit

Linux Sowirdsgemacht−Beschreibungen füllen die Lücke zwischen Büchern und kurzen Liesmich−Dateien. Sie haben den Anspruch, einen Sachverhalt ausführlich zu erklären. Eigens zu diesen Zweck wurde vom Linux−Dokumentations−Projekt (LDP) eine umfangreiche Infrastruktur aus Autoren, Schreibwerkzeugen und Distributions−Kanälen geschaffen. Das vorliegende Sowirdsgemacht ist offizieller Teil des (LDP) und somit wie alle anderen Dokumente des LPD erhältlich. Entweder auf dessen Webseiten oder im Verzeichnis /usr/share/doc/howto/ auf der eigenen Festplatte. Daneben sind gedruckte gedruckten Zusammenstellungen verschiedener Verlage über die Buchhandlungen erhältlich.

Die neueste Version des Deutschen Sowirdsgemacht ist ausserdem über meine hauseigene Seite erhältlich.

Korrekturen und Verbesserungsvorschläge können per E−Mail an mich geschickt werden: me@wt.xpilot.org. Angebote für mehr Schleichwerbung in der Einleitungen nehme ich dort ebenfalls entgegen.

Offene Punkte

- Korrekturlesen lassen. :)
- Danksagungen auf den neusten Stand bringen und wieder hinzufügen
- Quelltext in das neue docbook−Format konvertieren.

1.2 Menschen

Benutzergruppen

In vielen Städten haben sich Linux−Anwender zu sogenannten Linux User Groups (LUGs) zusammengetan. Diese Benutzergruppen sind offen für jeden und halten regelmäßige Sitzungen ab. Sie sind der ideale Treffpunkt für all diejenigen, die sich nicht nur im "Cyberspace" mit anderen Linuxern austauschen wollen. Eine übersicht befindet sich beispielsweise auf den deutschen Linux−Portalseiten.

Speziell für Frauen

Hm, ich habe bisher nur die KDE−Frauen gefunden. Gibt es noch mehr?

Vereine

Neben den lokalen LUGs gibt es auch noch die folgenden Vereine:

Der Linux−Verband richtet sich in erster Linie an Firmen−Mitglieder. Die Aktivitäten sind beachtlich und konzentrieren sich im Moment auf die Aufklärung über die Schädlichkeit von Software−Patenten.

Die German Unix User Group zielt nach Jahren der Neuorientierung derzeit wieder auf Unix−Systemadministratoren.

Kongresse

Derzeit gibt es drei grosse überregionale Veranstaltungen zu Linux in Deutschland: den LinuxTag, den Linux−Kongress und die LinuxWorld Expo. Darüber hinaus gibt es zahlreiche lokale Veranstaltungen, die in den Kalendern der Portal−Seiten zu finden sind.

Diskussionsforen

Erste Anlaufstelle für deutschsprachige Anwender ist die "Newsgroup" de.comp.os.unix.linux.infos. Dort sind weitergehende Informationen zu Linux und über die Benutzung der zugehörigen "Newsgroups" zusammengetragen. Eine gute Idee ist die Benutzung der web−basierten Archive, in denen man zu Stichworten eine Menge Antworten findet: Google−Suche: de.comp.os.linux.* oder Oft gestellte Fragen der Deutschen Linux−Diskussionsgruppen.

Internet Relay Chat (IRC)

Der IRC−Kanal #LinuxGER bietet eine Hotline zu Linux in deutscher Sprache. Über dessen hauseigene Web−Seite kann man mehr über die Welt des IRC lernen.

1.3 Dokumentation

Deutschsprachige Portalseiten zu Linux

Hier ist eine kleine Auswahl von Seiten, die schon länger existieren und auch regelmässig gepflegt werden:

Linux in Österreich

Linux in der Schweiz

Linux.de

Pro Linux

Deutsches Linux Howto Projekt (DLHP)

Das Deutsche Linux Howto Projekt (DLHP) Projekt beschäftigt sich mit der übersetzung der originalen englischsprachigen Linux Sowirdsgemacht ins Deutsche. Auf den Webseiten des Projekts kann man eine Liste aller übersetzten Sowirdsgemacht und nähere Informationen zum Projekt abrufen.

1.4 Zeichensätze

Rechner sind Geräte, die auf der Grundlage von digitalen Zahlen arbeiten. Menschen sind Wesen, die mit Zahlen und auch mit Namen arbeiten. Setzt man einen Menschen vor einen Rechner und verkauft dem Menschen den Rechner als Universalgerät, dann muss der Rechner neben den Zahlen auch mit Namen umgehen können. Dies wird mit einem Kunstgriff erreicht: Das Alfabet wird durchnummeriert, d.h. Buchstaben werden Zahlen zugeordnet, mit denen der Rechner wiederum umgehen kann. Beispiel: Der Buchstabe A hat die Nummer 65. Zeichentabelle oder Zeichensatz wird diese Zuordnung genannt. Sie listet neben den Buchstaben auch noch Symbole und Sonderzeichen auf. Der Zeichensatz ist eine rein logische Definition; erst die Schriftart bestimmt die Darstellung des Zeichensatzes auf dem Bildschirm.

Damit Dokumente zwischen Computern austauschbar sind, muss der verwendete Zeichensatz standardisiert sein. Ein früher Standard war US−ASCII, eine sehr kleine, aus 127 Zeichen bestehende Zuordnung, bei der jedes Zeichen genau 7 Bits Speicherplatz belegte. Ein Bit mehr pro Zeichen benötigt der Zeichensatz ISO−8859−1, mit dem sich die meisten mitteleuropäischen Sprachen darstellen lassen. Allerdings legen beide Zeichensätze das lateinische Alfabet zugrunde, sind also für Anwender ausserhalb von Europa oder Amerika nur begrenzt nützlich. So entstanden in den verschiedenen Regionen der Welt Zeichensätze, die wegen unterschiedlicher Zeichen mit derselben Nummer nicht miteinander verträglich sind.

Tabelle 1: Ausschnit des Zeichensatzes ISO−8859−1

```
"  quotedbl      '  apostrophe    `  grave         ~  asciitilde
^  asciicircum   _  underscore    !  exclam        ?  question
#  numbersign    $  dollar        %  percent       &  ampersand
|  bar           @  at            +  plus          −  minus
*  asterisk      /  slash         \  backslash     .  period
,  comma         :  colon         ;  semicolon     <  less
=  equal         >  greater       (  parenleft     )  parenright
[  bracketleft   ]  bracketright  {  braceleft     }  braceright

¡  exclamdown    ¢  cent          £  sterling      ¤  currency
¥  yen           ¦  brokenbar     §  section       "  diaeresis
©  copyright     ª  ordfeminine   <  guillemotleft
¬  notsign       −  hyphen        ®  registered
¯  macron        °  degree        ±  plusminus     ²  twosuperior
³  threesuperior                  ´  acute         µ  mu
¶  paragraph     ·  periodcentered                 ¸  cedilla
¹  onesuperior   º  masculine     >  guillemotright
¼  onequarter    ½  onehalf       ¾  threequarters
```

```
¿  questiondown À  Agrave       Á  Aacute       Â  Acircumflex
Ã  Atilde       ä  Adiaeresis   °  Aring        Æ  AE
Ç  Ccedilla     È  Egrave       É  Eacute       Ê  Ecircumflex
Ë  Ediaeresis   Ì  Igrave       Í  Iacute       Î  Icircumflex
Ï  Idiaeresis   Ð  ETH          Ñ  Ntilde       Ò  Ograve
Ó  Oacute       Ô  Ocircumflex  Õ  Otilde       ö  Odiaeresis
×  multiply     Ø  Ooblique     Ù  Ugrave       Ú  Uacute
Û  Ucircumflex  ü  Udiaeresis   Ý  Yacute       Þ  THORN
ß  ssharp       à  agrave       á  aacute       â  acircumflex
ã  atilde       ä  adiaeresis   å  aring        æ  ae
ç  ccedilla     è  egrave       é  eacute       ê  ecircumflex
ë  ediaeresis   ì  igrave       í  iacute       î  icircumflex
ï  idiaeresis   ð  eth          ñ  ntild        ò  ograve
ó  oacute       ô  ocircumflex  õ  otilde       ö  odiaeresis
÷  division     ø  oslash       ù  ugrave       ú  uacute
û  ucircumflex  ü  udiaeresis   ý  yacute       þ  thorn
ÿ  ydiaeresis
```

Die derzeitige Lösung ist der internationale Unicode−Zeichensatz (identisch zu ISO−10646). Unicode steht für unique code und bezeichnet eindeutige Nummern von derzeit über 90.000 definierten Zeichen in einer einzigen Zuordnungstabelle. Der Prozess der Standardisierung ist weit fortgeschritten, aber noch nicht abgeschlossen. Schätzungen gehen davon aus, dass niemals mehr als ca. 2.1 Millionen (21 Bit) Zeichen definiert sein werden. Um eine ausreichende Reserve vorzuhalten, wurden die Unicode−Tabelle mit 32 Bit pro Zeichen ausgelegt, was zur Nummerierung von mehr als 4 Millarden Zeichen ausreicht. Unicode bricht also mit einer geläufigen Annahme: 8 Bit (ein Byte) pro Zeichen. Die vier Byte scheinen grosszügig dimensioniert und werfen Fragen des Speicherverbrauchs auf: Mit unserem mitteleuropäischen Zeichensatz ISO−8859−1 belegt jedes Zeichen acht Bit auf den Speichermedien (RAM, CD−ROM, Festplatte). Ohne spezielle Massnahmen belegen in Unicode dieselbem Zeichen 32 Bit, d.h. alle Speichermedien müssten für die Verwendung von Unicode vier mal so gross ausgelegt werden. Das erschien nicht durchführbar, weshalb zur effektiveren Speicherung verschiedene Unicode Bit Transformationen (UTF) eingeführt wurden: UTF−8, UTF−16 und UTF−32. Letztere Transformation ändert nichts, spart deswegen auch nichts und belegt somit in verschwenderischer Manier für jedes Zeichen konstant 32 Bit im Speicher. Erstere Transformationen (UTF−8) belegt zwischen acht und 48 Bits, je nach Standort des Zeichens in der Tabelle. Oft benötigte Zeichen (US−ASCII) stehen vorne in der Tabelle und belegen nur acht Bits, die deutschen Umlaute stehen mehr zur Mitte hin und belegen 16 Bits. Ein kleiner Nachteil von UTF−8 ist die rechenzeitintensive Transformation der Daten. Die Transformation UTF−16 liegt irgendwo zwischen dem Speicherplatzverbrauch von UTF−32 und dem Rechenzeitverbrauch von UTF−8.

Weitere Informationen zu Unicode sind direkt beim Unicode−Consortium und in der UTF−8 and Unicode FAQ for Unix/Linux (engl.) zu finden.

Das Euro−Symbol

Das ISO Währungs−Kürzel für den Euro ist EUR, wie man es von den Überweisungs−Formularen der Banken schon kennt. Die Schreibweise EUR 10,23 für 10 Euro und 23 Cent ist gleichberechtigt mit der Verwendung des speziellen Währungs−Symbols ¤ (E mit zusätzlichem Mittelstrich). Falls das Euro−Symbol eher wie ein mechanischer Wecker aussieht, dann verwenden Sie noch die alten Schriftarten. In diesem Fall sollten Sie Ihre Linux−Distribution aktualisieren, denn ein Austausch aller denkbaren Schriftarten geht weit über die Möglichkeiten dieses Sowirdsgemacht hinaus. (Schriftarten waren schon immer ein Problem unter Linux, was das Euro−Symbol nur bestätigt.) Ob sich eine Aktualisierung angesichts des robusteren ISO−Kürzels EUR überhaupt lohnt, müssen Sie selbst entscheiden.

Definiert wird das Euro−Symbol vom Zeichensatz ISO−8859−15. Es liegt in der Tabelle bei den anderen Währungs−Symbolen zwischen Pfund und Yen. Über die Tastatur können Sie es wie aufgedruckt mit

AltGr−E eingeben. Die Tastenkombination AltGr−C gibt das bekannte Cent−Symbol.

1.5 Tastatur

Alle dem Autor bekannten Distributionen stellen die deutsche Tastaturbelegung selbstständig bei der Installation ein. Ist man mit dem Resultat nicht zufrieden, so lassen sich mit den Kommandozeilen−Werkzeugen loadkeys (Text−Konsolen) und xmodmap (X11) andere Belegungen einstellen. Eine genauere Beschreibung der Werkzeuge geht über den Anspruch dieses Dokuments hinaus.

Bei dieser Gelegenheit soll ein wenig Licht in die Konzepte der Tastaturbelegung gebracht werden, um unter Unix/Linux in Zukunft weniger Probleme mit den Tastenbelegungen zu haben. Die Betätigung einer Taste sendet eine Kennzahl. Andere Tastaturen geben andere Kennzahlen. Ähnlich zu den weiter oben beschriebenen Zeichensätzen, kommen auch bei der Tastatur Zuordnungstabellen zum Einsatz, die den Kennzahlen Symbole zuordnen. Die Zuordnungstabellen heissen Tastaturbelegungen. Eine Tastaturbelegung definiert unter Umständen nicht direkt die auszugebenden Zeichen, sondern machen einen Zwischenschritt über symbolische Namen. Beispiel: Unter X11 sendet eine Taste die Kennzahl 22. Die Tastaturbelegung definiert für diese Kennzahl (Terminus "keycode") das Symbol BackSpace, womit die Taste zum Löschen des linken Zeichens auf der PC−Tastatur gemeint ist. X11−Anwendungen reagieren direkt auf dieses Tasten−Symbol. Für Anwendungen, die in einem Textdialog−Fenster (xterm, ssh) ablaufen, wird das Symbol widerum in eine Steuersequenz übersetzt, z.B. Steuerung−? oder Escape−[3 .

Kompositionen

Gebräuchliche Tastaturen verfügen über ca. 110 Tasten. Um damit den kompletten Zeichensatz ISO−8859−1 eingeben zu können, müsste man die Tasten mit bis zu vier Zeichen belegen. Benötigt man nur selten den Zugriff auf den kompletten Zeichensatz, dann ist die Vierfach−Belegung ein zu hoher Lernaufwand. In diesem Fall bietet sich die Kompositions−Taste an. Diese Taste verursacht selbst keine Ausgabe auf dem Bildschirm, sondern komponiert die beiden nachfolgend eingegebenen Zeichen zu einem. Mit anderen Worten: Die Zeichen werden übereinander gedruckt. Beispiel: Die Tastenfolge <Compose> <~> <a> ergibt ein ã. Unter Linux ist meist eine der Tasten rechts von der Leertaste mit der Kompositions−Funktion belegt. (Hinweis für Xmodmap−Bastler: Das XFree−Tasten−Symbol der Kompositions−Taste ist Multi_key.)

Ständige Komposition

Bestimmte Tasten können als ständig komponierend konfiguriert werden. Die Betätigung solcher Tasten führt zunächst zu keiner Ausgabe auf dem Bildschirm, sondern das betreffende Symbol wird über das nächste Zeichen gedruckt, das eingetippt wird.

Die Funktion wird im Englischen manchmal mit "dead keys" bezeichnet, was eine Fehlbezeichnung ist, denn tote Tasten verursachen überhaupt keine Funktion.

Das Bootprompt

Beim Starten des Computers wird der Linux−Kernel von einem sogenannten Boot−Loader in den Arbeitspeicher geladen. Über den Boot−Loader kann man dem Kernel Parameter übergeben − vorausgesetzt, man kennt die amerikanische Tastenbelegung. Die folgenden Abschnitte beschreiben die Möglichkeiten zur Konfiguration der Tastenbelegung für die Boot−Loader von Linux.

Lilo

Seit der Version 2.0 hat die ehrenwerte Dame die Fähigkeit zur Verwendung alternativer Tastenbelegungen. Dem Source−Code liegt das Perl−Skript keytab−lilo.pl bei, mit der man eine der Tastaturbelegungen für die Textkonsolen in für Lilo geeignetes Format überführt. Beispiel: man möchte die Tastenbelegung de−latin1−nodeadkeys.map schon auf dem Lilo−Prompt geladen wissen. Dazu ist folgender Befehl auszuführen:

```
./keytab-lilo.pl  de-latin1-nodeadkeys  > /boot/de-latin1-nodeadkeys.ktl
```

Um dises Belegung zu aktivieren muß man in der Konfigurationsdatei /etc/lilo.conf folgenden Eintrag vornehmen und danach das Kommando lilo ausführen. Die Details können sich allerdings von Distribution zu Distribution unterscheiden, weshalb man bitte die mitgelieferten Handbücher konsultiere.

```
keytable = /boot/de-latin1-nodeadkeys.klt
```

1.6 Zeitzone

Die Erde ist in Zeitzonen eingeteilt, innerhalb derer die Tageszeit von der natürlichen Zeit (höchster Sonnenstand = Mittag) abweichen kann. Die Tageszeit der Zeitzone kann sogar springen, wie beispielsweise bei der Umstellung zwischen Sommer und Winterzeit. Bei den Zeitzonen handelt es sich also um politische Festlegungen. Für kleine Länder kann man die Zeitzone über die Hauptstadt auswählen, also Europe/Berlin, Europe/Vienna oder Europe/Zurich. Hinter diesem Alias−Namen verbirgt sich die tatsächliche Zeitzone, nämlich in unserem Fall die Mitteleuropäische Zeit (Central European Time, CET) bzw. die Mitteleuropäische Sommerzeit (Central European Summer Time, CEST).

Das früher gebräuchliche MET sollte unter keinen Umständen weiterverwendet werden. Von Markus Kuhn stammt folgende Information zu diesem Thema: Die Datei MET existiert nur zwecks Rückwärtskompatibilität. Die Autoren der Zeitzonentabellen (Olson/Eggert/et al.) wollen eigentlich, daß deutsche Anwender statt MET die Datei Europe/Berlin verwenden. Dann stimmen sogar die historischen Sommerzeiten vor 1945, die Linux auch alle kennt. Außerdem wird dann in der neuesten Version der Zeitzonentabelle "Mitteleuropäische Zeit (MEZ)" endlich korrekt ins Englische mit "Central European Time (CET)" übersetzt, denn "MET" ist ein übersetzungsfehler (siehe z.B. Langenscheid Englisch). Ich habe deswegen sogar beim Physikalisch−Technischen Bundesamt nachgefragt, die für die deutsche Zeit verantwortlich sind.

Der gemeinsame Bezugspunkt für alle Zeitzonen ist die Universalzeit (Universal Time, UTC), früher auch Greenwich Mean Time (GMT) genannt. Unter Linux lässt man die CMOS−Uhr auf der Hauptplatine üblicherweise mit UTC laufen. Der Linux−Kernel übernimmt den Wert der CMOS−Uhr beim Systemstart und errechnet daraus die Sekunden seit dem 1. Januar 1970, 0 Uhr (Unix Epoch Time). Intern arbeitet der Linux−Kernel nur mit dieser Sekunden−Zahl, die üblicherweise auch in UTC läuft. Soll dem Anwender ein Datum oder eine Zeit angezeigt werden, dann wird aus der Systemzeit unter Berücksichtigung der Zeitzone und der Lokalisierung (siehe oben) eine Zeichenfolge konstruiert.

Zusammenfassend kann man sagen, dass es unter Linux 3 Zeiten gibt: Die der auf dem Motherboard installierten CMOS−Uhr, die Systemzeit des Linux−Kernels und die dem Anwender von verschiedenen Uhren angezeigte Zeit. Die Kommandozeilen−Werkzeuge hwclock ("hardware clock") und date übersetzen zwischen diesen Uhrzeiten. Folgende Strichzeichnung verbildlicht die Zusammenhänge:

```
                hwclock −u −w              date −u −s
     CMOS−Uhr  <−−−−−−−−−−−−−−   Linux    <−−−−−−−−−−−−−−
```

```
    (UTC)                    Systemzeit                 Anwender−Uhr
             −−−−−−−−−−−−−−>  (Epoch)    −−−−−−−−−−−−−−>
              hwclock −u −s               date, xclock
```

Zwar liest der Kernel schon beim Booten die CMOS−Uhr aus und interpretiert sie als UTC. Aber um systematische Fehler der CMOS−Uhr zu korrigieren, wird die Uhrzeit von einem der Systemstart−Skripte im Verzeichnis /etc/init.d/ ein weiteres Mal gelesen. Übernimmt man die Linux−Systemzeit mit dem Kommando netdate von einem Zeit−Server, dann muss man die CMOS−Uhr mit dem Kommando hwclock nachziehen.

Die systemweit gewählte Zeitzone kann mit der Umgebungsvariable TZ individuell überschrieben werden. Beispiele:

```
export TZ=Asia/Hong_Kong; xclock &
export TZ=UTC; xclock &
```

1.7 Anwendungen

Internationalisierung und Lokalisierung

Internationalisierung und Lokalisierung sind auch im Englischen lange Wörter, weshalb sie mit i18n und i10n abgekürzt werden. Die Zahlen geben an, wieviele Buchstaben in der englischen Schreibweise ausgelassen wurden.

Mit i18n wird die Veränderung eines Programms zur Unterstützung mehrerer Sprachen bezeichnet. Bei diesem Prozess wird normalerweise der angezeigte Text (Menüs, Meldungen, usw.) vom Programmcode getrennt. Auf diese Weise können sich mehrere Übersetzungen denselben sprachenunabhängigen Programmcodes teilen. Ins Deutsche übersetzte Programmeldungen können über die Umgebungsvariable LANG ausgewählt bzw. aktiviert werden. Beispiel:

```
#LANG=de_AT  # in österreich
#LANG=de_CH  # in der deutschsprachigen Schweiz
LANG=de_DE   # in Deutschland
export LANG
```

Das Resultat ist nicht immer zufriedenstellend. Beispielsweise sind nicht alle Bibliotheken internationalisiert, so dass sich die Ausgabe−Texte mit Englisch mischen. Die Entwicklung in den vergangenen Jahren lässt sich an folgendem Beispiel ablesen:

```
# im Jahre 1997:
tar: Kann Archiv 'foo.tgz' nicht öffnen: Permission denied

# im Jahre 2001:
tar: foo.tgz: Cannot open: Keine Berechtigung
```

i10n bezeichnet die darüber hinaus gehende Feinarbeit der Anpassung an nationalen Besonderheiten, wie Datums− und Zahlenformate oder das kulturelle Umfeld. Eine deutsche Lokalisierung wird mittlerweile automatisch mit der LANG−Umgebungsvariable aktiviert. So macht es am meisten Sinn. Bei älteren Systemen kann mit der Umgebungsvariable LC_ALL die Lokalisierung eingeschaltet werden. Beispiel:

```
LC_ALL=de
export LC_ALL
```

Die Einstellungen sollten von allen modernen Linux−Distributionen automatisch vorgenommen werden. Exotischere Lokalisierungen sind zum Beispiel von IBM erhältlich: IBM developer works: universal locales

Ausnahmen von der Regel

Sowohl für die Textkonsole als auch für X11 existieren Zeichensätze, die nicht alle Zeichen aus ISO−8859−1 enthalten. Dies gilt erst recht für Unicode. Die fehlenden Zeichen (z.B. Umlaute) werden als Leerzeichen oder gar nicht auf dem Bildschirm dargestellt. In diesem Fall hilft natürlich die nachfolgend beschriebene Konfiguration der Anwendungen nicht weiter, sondern es muß ein anderer (vollständiger) Zeichensatz gewählt werden.

```
Anwendung    Einstellung                vorgenommen in/auf
-------------------------------------------------------------------
LaTex        \usepackage{a4}            im Dokument
gnuplot      set encoding iso_8859_1    ~/.gnuplotrc
xfig         Fig*inches:   false        Xresource
xfig         -metric                    Kommandozeilenoption
lynx         CHARACTER_SET:iso-8859-1   ~/.lynxrc und /etc/lynx.cfg
nn                                      ~/.nn/init und /etc/nn/setup
             set data-bits 8
             set charset iso-8859-1
tin                                     ~/.tin/headers
             Mime-Version: 1.0
             Content-Type: text/plain; charset=iso-8859-1
             Content-Transfer-Encoding: 8bit
elm                                     ~/.elm/elmrc und /usr/lib/elm/elm.rc
             charset = iso-8859-1
             displaycharset = iso-8859-1
             textencoding = 8bit
pine         character-set=ISO-8859-1   ~/.pinerc und /etc/pine.conf
less         LESSCHARSET=latin1         Umgebungsvariable
joe          -asis                      ~/.joerc und /etc/joe/joerc
dosemu                                  ~/.dosrc und /etc/dosemu.conf
             keyboard { layout de-latin1  keybint on  rawkeyboard on }
             X { updatefreq 8 title `DOS in a BOX' icon_name `xdos' keycode }
kermit                                  ~/.mykermrc und /etc/kermit.ini
             set terminal bytesize 8
             set command bytesize 8
             set file bytesize 8
             set language german
             set file character-set latin1-iso
             set transfer character-set latin1-iso
             set terminal character-set latin1-iso
```

Texte

Das Papierformat DIN A4

In Amerika wird hauptsächlich das Papierformat US−Letter verwendet, welches ein wenig breiter und niedriger ist als das in Europa gebräuchliche DIN A4. Oft muss man Anwendungen ausdrücklich auf DIN A4 einstellen. Beispiele:

```
Anwendung    Einstellung                vorgenommen in/auf
-------------------------------------------------------------------
LaTex        \usepackage{a4}            im Dokument
xdvi         XDvi.paper: a4             X-Resource
ghostview    Ghostview.pageMedia: A4    X-Resource
```

```
ghostscript  -sPAPERSIZE=a4           Kommandozeile
dvips                                 Konfigurationsdatei .dvipsrc
             @ a4size 210mm 297mm
             @+ %%PaperSize: a4
ImageMagick  -page A4                 Kommandozeile
```

Austausch von Texten mit anderen Systemen

Die Textformate von DOS/Windows−basierten System, Apple−Computern und UNIX unterscheiden sich in zwei Punkten: Zeichensatz und Zeilenenden. Linux macht am Ende einer Textzeile nur einen Zeilenvorschub, Äpfel machen nur einen Wagenvorlauf und DOS−basierte Systeme machen beides.

Die Programme `fromdos`, `todos` und `frommac`, `tomac` wandeln ausschließlich die Zeilenenden um, die Zeichen selbst werden nicht angepaßt. Gleiches gilt für die Optionen `conv=auto', bzw. `conv=text' des mount−Kommandos bezogen auf das MS−DOS Dateisystem.

Will man auch die Zeichensätze von Texten konvertieren, dann hilft das Kommandozeilen−Werkezeug recode weiter. Durch die Definition von Aliasen kann man sich das Leben etwas einfacher machen:

```
alias unix2dos='recode lat1.ibmpc'
alias dos2unix='recode ibmpc.lat1'
alias unix2mac='recode lat1.mac'
alias mac2unix='recode mac.lat1'
```

Drucken

Die folgenden Überlegungen betreffen nur Text−Dateien. Grafiken oder die Ausgabe von Textverarbeitungen werden als Punktmuster zum Drucker geschickt und sind daher nicht betroffen.

Gängige Drucker−Modelle erwarten Texte im DOS−Format. Schickt man eine Textdatei im Unix−Format zum Drucker, dann erhält man einen Treppenstufen−Effekt, weil der Text am Ende der Zeile keinen Wagenrücklauf aufweist. Beispiel: lpr /etc/nsswitch.conf. Entweder konvertiert man Textdateien wie oben beschrieben in das DOS−Format oder man wandelt sie mit Werkzeugen wie atp, mpage, a2ps, ascii2ps, encode oder gencode in das PostScript−Format. Für PostScript installieren die modernen Linux−Distributionen Drucker−Treiber.

---

2. English Part (Englischer Teil)

2.1 Introduction

The Linux German Howto explains how to enable German specific features for Linux applications as well as for the Linux system. But what is German specific? If you have been to the worlds largest computer exhibition CeBIT in Hannover, you may consider traditional leather costumes, white sausages, (real!) beer and perfectely ironed underwear to be typical for germany. Please beg my pardon that I can't comment on the underwear part or else this Howto would at least get a R−rating, which would trigger questions about distribution restrictions and their impacts for free software and OpenSource and OpenDocumentation and so on. In the end we will surely have an OpenRating as well. Pure horror.

Regarding the other three points about germany mentioned above, these are correct as you may already have figured out yourself. Believe me, I'm just wearing such a traditional costume while typing the text in front of his computer. Everybody does all the time. Even my computer wears a leather costume, or else it would not be germany! Built into the leather case is a sound microprocessor to turn the simple PC−beep into an original south german yodel. Can you already see the alp mountains?

So what else is german specific? Maybe your friendly stewardess told you the waiter at a restaurant "is not offensive but doing regular service to you" before your plane arrived at the german airport. Cultures clashing into each other. Of course the waiter is doing his best, but at least the americans are not willing to understand. Example: In my home town the waiters are employed to serve the beer. This is a fundamental different concept than just aiming at bloody customer satisfaction. It counts that the beer can flow down your throat in a fresh state. The beer must be satisfied. Think of it as a religious believe. Next time the american president secretly visits me, I will explain and show him all the details. Promised.

Also quite typical for German are long words and long sentences. Example: The German translation of Howto is Sowirdsgemacht. More than double as long. My words are longer than yours. Ha! We can even go for extremes like in Dampfschifffahrtspensionskassenchefsekretärinnenhalbjahresbetriebsausflugsbudget. The typsetter will surely hate me for using such a long word. D78g, now all−together: Nyia−hahaha. As you can imagine, most applications will die because of string buffer overflows when german users start to type in their unusal long words. That is what the German Howto is about.

Not to forget our German speaking friends at the south: Austria and Switzerland. Read this Howto and learn how to manage your secret swiss bank account from within Linux. Making money fast is twice as much fun then.

Well, before I completely destroy the image of over−serious germans, lets come to the point. Linux is developed by a world wide community of hackers on the internet. Their least common denominator for communication is English and that works quite acceptable. The situation for users is different. In order to do their work, they have to use the native language and the specifics of their country. Examples: characters, time zones and numbering conventions. Luckily Linux is so widespread, that national markets have already become a target for Linux distributions. In other words, most adaptions are already built in. In this context the KDE project should be mentioned, as it surely has set milestones for the amount, ease and completeness of internationalization in applications. The German Howto honors this evolution by concentrating on basic knowledge instead of endless parameter listings.

The English part of the Howto is much shorter than the German one. It is only inteded as an overview of the situation.

2.2 Copyright, licence and disclaimer

Short: use this Howto at your own risk.

2.3 Availability

Linux Howtos fill the gap between books and short readme files. They explain one subject in a detailed fashion. To fullfill this goal, the Linux Documentation Project (LDP) has created a infrastructure consisting of authors, tools and distribution channels. The Linux German Howto is part of the LDP and thus available like all other Howtos. Either on the LDP homepage or under the directory /usr/share/doc/howto/ on a typical Linux installation. Printed collections of Howtos published by various companies are available at your local bookstore.

The latest version of this document can be downloaded from my homepage.

Corrections and suggestions should be sent via email to my address me@wt.xpilot.org. Paid contracts for product placements in the introduction are also welcome.

2.4 Linux in German speaking countries

This is a small list of Linux portals, which are maintained and updated frequently:

Linux in Österreich

Linux in der Schweiz

Linux.de

Pro Linux

A large number of local Linux User Groups exists in Austria, Germany and Switzerland. You can find them in the directories of most German Linux Portals. Country wide associations are the Linux−Verband, which aims more at commercial members, and the German Unix User Group (GUUG), which aims at system administrators. Three large conventions are held each year, the LinuxTag (LinuxDay), the Linux−Kongress and the LinuxWorld Expo. Local events are usally organised by the Linux User Groups and can be found in the calendars. The newsgroup hierarchy de.comp.os.unix.linux.* is a quite busy part of the german usenet. The same is true for the IRC channels #debian.de and #linuxger.

2.5 Configuration

Characters

The traditional character sets for central europe are ISO−8859−1 and ISO−8859−15 (including the euro symbol). Please see the german part of this Howto for an excerpt of ISO−8859−15. Unicode covers both sets, but only few users have applications with full support for unicode. That may change in the future.

Conversions of the character sets have also be done when exchanging texts between Unix/Linux, Macintoshs and DOS/Windows systems. It is not enough to just convert the line endings. This is also true for printing; most printers expect the DOS−style text format. (When printing from word processors and such, high quality binary bitmaps are sent to the printer, which are not affected by character set and line ending problems.)

Quite annoying is the fact that y and z are exchanged compared to an english keyboard. Not to speak of the special characters like the slash. Now imagine you want to type yes and it always gives zes. Or you want to type /dev/sda and it gives ?dev?sda. Short: a keyboard mapping is required in europe. Every european country has its own keyboard layout. The right mapping can be choosen at installation time for all modern Linux distributions. Suitable for germany and austria is de−latin1−nodeadkeys, whereas the swiss people need sf−latin1 (swiss−french) or sg−latin1 (swiss−german).

Please keep in mind that the loadkeys command only helps for a properly started system. To have the correct mapping even for the Lilo boot prompt, you have to create a key mapping with keytab−lilo.pl and configure Lilo to use that mapping.

The character set ISO−8859−1 shares the first 128 characters with US−ASCII and defines another 128 characters on top of it. To type in all these characters on the keyboard, it would be necassary to map up to four symbols per key. For cases where you access the complete set only occasionally,such a mapping would be a too high learning effort. One solution is the use of the compose key. After pressing the compose key, which is usally mapped to the right control key, the next character is not displayed but instead printed over the following character. Example: the sequence Right−Ctrl   A is composed as the character Ã (capital a with a tilde on top).

An alternative is the feature of always composing keys. Under such mappings, characters like   are never displayed but always printed over the following character. Usally this feature is refered to as dead keys, which is a little bit misleading. I'm not aware of anybody using the dead keys feature.

Currency

On january 1st 2002, the last step of the currency union in europe was performed. Since then the currency is euro and its fraction is cent. Although there is a special currency symbol in ISO−8859−15 and in unicode, you can safely use the ISO currency code EUR for euro as you used ATS for austrian schillings or DEM for deutsche marks before. Please note that switzerland (CHF, swiss francs) is not part of the european union.

XFree86

There is nothing special about XFree86 for German users. Howevery, several hardware manufactures asked me for the correct setting, which is why I'm citing them here.

```
# for XFree86 4.1.* only:
Section "InputDevice"
  Driver       "Keyboard"
  Identifier   "Keyboard[0]"
  Option       "Protocol" "Standard"
  Option       "XkbLayout" "de"
  Option       "XkbModel" "pc104"
  Option       "XkbRules" "xfree86"
  Option       "XkbVariant" "nodeadkeys"
EndSection

# for XFree86 3.* only:
Section "Keyboard"
  Protocol      "Standard"
  XkbRules      "xfree86"
  XkbKeycodes   "xfree86"
  XkbModel      "pc104"
  XkbLayout     "de"
  XkbVariant    "nodeadkeys"
```

```
EndSection
```

2.6 Time zone

The time zones for central europe are CET (Central European Time) and CEST (Central European Summer Time) respectively. However, system administrators are expected to use one of the aliases Europe/Berlin, Europe/Vienna or Europe/Zurich. This gives correct results even before the Unix Epoch.

Internationalization and lokalization

Internationalization and lokalization are unusal long words (but not as long as D78g, see above) and thus abreviated by i18n and i10n. The numbers indicate how many characters have been left out.

I18n denotes the changes to the program code in order to have a multi lingual output. A common implementation is to seperate the messages from the code. This way several translated texts share the same code base. German messages are activated with the LANG environment variable. Examples:

```
#LANG=de_AT  # for austria
#LANG=de_CH  # for switzerland (German)
LANG=de_DE   # for germany
export LANG
```

The result is not always satisfying, because not all applications and libraries are internationalized. It may happen that German and English messages are mixed like in the following example, which also illustrates the evolution of the problem:

```
# in 1997:
tar: Kann Archiv 'foo.tgz' nicht öffnen: Permission denied

# in 2001:
tar: foo.tgz: Cannot open: Keine Berechtigung
```

German messages alone don't make much sense. Applications must also display the data using the correct format. Example: 08.07.2001 instead of 07.08.2001 or EUR 10.000,32 instead of EUR 10,000.32. You get the idea. The required changes to the programm are denoted as i10n. Today the LANG variable also activates the associated locale.

Paper size

The standard paper size in europe is DIN A4, which is a little bit smaller and taller than US−letter. The aspect ratio of height vs. width is the square root of two. All sizes are derived from A0 by a centered cut at the longer side.

---

3. Fun für Dich: Denglish

Das erste Kapitel in Deutsch, das zweite in Englisch. Was liegt näher, als im Dritten beide Sprachen zu mischen? Zur genaueren Erörterung dieser Frage möchte ich an dieser Stelle ein Beispiel aus der Welt der Programmiersprachen geben. Programmiersprachen haben den Vorteil, dass sie einen einfachen Grund−Wortschatz aufweisen und nicht gesprochen werden, d.h. reine Schriftsprachen sind. Vorranging um Schriftsprache geht es ja bei der einleitenden Frage.

```
# Perl und Bourne gemischt:
system("rm wochenplan.pdf");

# natives Perl
unlink("wochenplan.pdf");
```

Auch Programmiersprachen lassen sich mischen, wie im Beispiel gezeigt wird. Die Perl−Funktion system ruft den Bourne−Interpreter auf, der seinerseits das externe Programm rm zur Entfernung der Datei geschäftsbericht.txt aufruft. Eine Recht komplizierte Vorgehensweise, welche aber oft in Computer−Programmen von Anfängern zu finden ist, denen noch die Erfahrung im Umgang mit der Sprache fehlt. Das fehlende Wissen über die Funktion unlink wird kompensiert durch die Kombination der Universal−Funktion system mit dem Wissen über das externe Werkzeug rm zur Entfernung von Dateien. Auf diese Weise ist das Programm in kürzester Zeit lauffähig.

Jeder erfahrene Perl−Programmierer wird bei der oben gezeigten Mischung der Sprachen einen Schrei des Entsetzens ausstossen, weil er um die damit verbundenen Gefahren weiss. Die Bedenken sind technischer Art, zum Beispiel könnten sich rm oder der Bourne−Interpreter später anders verhalten, schliesslich sind beide nicht Teil von Perl als Sprache. Als Folge könnte die Datei nicht oder eine andere Datei gelöscht werden. Und weil das in der Praxis tatsächlich vorkommt, wird immer wieder geraten, die Möglichkeiten einer Programmiersprache auszuschöpfen, bevor man andere Sprachen hinzumischt.

Leider ist es in den letzten Jahren üblich geworden, viel Englisch in die Deutsche Sprache zu mischen, obwohl dazu keine Notwendigkeit besteht. Aber bevor ich auf die möglichen Ursachen eingehe, ist es noch notwendig, den Bezug zu den Programmiersprachen herzustellen. Die deutsche Schriftsprache ist sicherlich sehr viel komplexer als Perl. Trotzdem ist der Zweck in beiden Fällen gleich: Informationen zu übermitteln, ohne dass deren Urheber zugegen wäre. Bei einem Buch werden in Abwesenheit des Autors Informationen an den Leser übermittelt, bei einem Programmtext an den Prozessor des Rechners. Genau das ist der Unterschied zur gesprochenen Sprache, bei der immerhin eine unterstützende Betonung gegeben werden kann, wenn nicht sogar der Autor persönlich zugegen ist und auf diese Weise Gesten oder Rückfragen möglich sind. In diesem Sinne gehört auch eine eingetippte Plauderei (engl. chat) im IRC, den virtuellen Räumen oder im Heise Newsticker trotz der Verwendung von Buchstaben eher zur gesprochenen Sprache.

Man kann also festhalten, dass es für die Informationsübermittlung Medien mit verschiedenen Bandbreiten gibt. Papier ist sozusagen das alte Modem, getippte Plaudereien sind schon DSL und die Anwesenheit vor Ort ist 100base−T Ethernet. Der Rückschluss "Im Chat haben mich immer alle verstanden, daher brauche ich zum Schreiben eines Buch keine Grammatik−Kenntnisse" führt daher nicht zum Erfolg: Was bleibt von einem Video übrig, das von DSL−Qualität auf Modem−Bandbreite herunterproduziert wird? Ich will an dieser Stelle nicht den vielzitierten Untergang des Abendlandes heraufbeschwören, aber die Notwendigkeit und die Vorteile von Schriftsprache für unsere Kultur sollte man schon verständlich vermitteln. Eine immer komplexer werdende Welt erfasst man besser mit einer guten Schriftsprache.

Drei Ursachen für die Verenglischung der Deutschen Sprache stellen sich mir dar:

- Künstliche Aufwertung von gewöhnlichen Produkten durch Verwendung der englischen Bezeichnungen. Beispiele: LCD−Display statt LCD−Anzeige, Music Awards statt Musikpreise, Server−Cluster statt Server−Gruppe oder Open−Air−Konzert statt Freiluftkonzert.
- Die Bequemlichkeit, die Kenntnisse aus Sprachen unreflektiert in andere zu übernehmen. (Siehe oben.)
- Fehlender Mut zu mehr Selbstbewusstsein. Das steigert sich dann bis hin zur Mystifizierung, man könne to mount nicht mit montieren übersetzen, weil noch andere (unerforschte?) Bedeutungen in "mount" stecken würden.

Natürlich gibt es auch den umgekehrten Fall, nämlich dass deutsche Worte hin und wieder in der englischen Sprache verwendet werden. Beispiele: Kindergarden, Autobahn oder heimlich. Aber dieser Anteil ist ohne Bedeutung und bleibt tatsächlich im Rahmen einer Bereicherung der Sprache, nicht der einer Verdrängung.

Die derzeitige Verenglischung des Deutschen halte ich aus den skizzierten Gründen für bedenklich, aber nicht für unausweichlich. Einzelne gewaltsam übersetzte IBM−Handbücher vergangener Tage sind zumindest für mich kein Argument, nicht nach besseren Lösungen zu suchen.
