

# Hinweise zum sinnvollen Umgang mit dem Computer

## Teil 1 — Kommandozeile und Netzwerk

Carsten Lemmen\*      Tobias Sperr

October 11, 2000

### 1 Die Kommandozeile unter Windows

Die Kommandozeile heißt im Allgemeinen *MSDOS-Eingabeaufforderung* oder nur *Eingabeaufforderung*. Dieses Programm kann auch gestartet werden, indem man die Datei `command.com` (unter Windows9x) oder `cmd.exe` (unter NT) ausführt.

**cd** Verzeichnis wechseln

Syntax `cd VERZEICHNISNAME`, ein übergeordnetes Verzeichnis erreicht man mit `cd .`, das Stammverzeichnis mit `cd \`

**dir** Verzeichnisinhalt anzeigen

**mkdir, rmdir** Verzeichnisse anlegen und löschen

**copy, delete** Dateien kopieren und löschen Syntax `copy QUELLDATEI ZIEL-DATEI`, um eine Datei in das aktuelle Verzeichnis zu kopieren, benutzt man `copy QUELLDATEI .`

**help, /?** Hilfe zu Befehlen anzeigen.

`help` ist unter Windows98 nicht mehr vorhanden. Jedem Befehl kann aber ein `/?` angehängt werden, man erhält dann eine kurze Hilfe.

---

\*[mail@carsten-lemmen.de](mailto:mail@carsten-lemmen.de)

## 2 Die Kommandozeile unter Unix

Die Kommandozeile wird im Allgemeinen unter Unix und Linux über einen Button auf dem Desktop gestartet. Das Eingabefenster wird dort als *Terminal* bezeichnet

**pwd** Wo bin ich?

Der Befehl `who am i` liefert die Antwort auf: "Wer bin ich?"

**cd** Verzeichnis wechseln (s.o.)

**ls, ls -al** Verzeichnisinhalt anzeigen.

Das DOS-Kommando `dir` funktioniert auf den meisten Unix-Systemen nicht (oder ergibt lustige Kommentare). Die Option `-al` hinter dem Befehl bewirkt, dass auch versteckte Dateien angezeigt `-a` und ausführlich dargestellt `-l` werden.

**mkdir, rmdir** Verzeichnisse anlegen und löschen.

**cp, rm** Dateien Kopieren und löschen.

**man, --help** Hilfe zu Befehlen anzeigen.

Die MANPAGES sind eine umfangreiche Hilfe. Innerhalb dieser bewegt man sich mit der Leertaste weiter, beendet wird die Hilfe durch `q`. Die Option `--help` wird dem fraglichen Befehl angehängt und bewirkt meist einen kurzen Hilfetext.

**ps, kill** Prozesse anzeigen und abbrechen.

**chmod** Benutzerrechte vergeben

Rechte sind Lesen (`r`, Wert 4), Schreiben (`w`, Wert 2) und Ausführen (`x`, Wert 1). Diese Rechte kann der Eigentümer, die Gruppe oder Andere haben. Beispielsweise bedeutet das Dateiattribut `-rwxr-x--`, dass der Eigentümer lesen, schreiben und ausführen darf, die Gruppe lesen und ausführen und andere gar nichts dürfen. Eine übliche Anweisung für Programmdateien ist `chmod 755 public_html`. Damit läßt man den Eigentümer alles tun ( $4+2+1$ ) und die Gruppe/Andere lesen und ausführen ( $4+1$ ).

## 3 Verbindungen zwischen Computern

### 3.1 File Transfer Protokoll — FTP

FTP ist in (fast) jedem Windows und Unix System zu finden. Man startet FTP mit dem Befehl `ftp` über die Kommandozeile. Darüber hinaus kann man beispielsweise den Internet Explorer (eingeschränkt) oder Programme wie WSFTP (Win) oder NCFTP (Win und Unix) nutzen.

Verbindung zu einem anderen Computer stellt man mit dem Befehl `open COMPUTERNAME` her:

```
open hpux11.hrz.uni-oldenburg.de
```

Man wird dann aufgefordert, den Benutzernamen und Passwort anzugeben. Sollte das scheitern, kann man sich mit dem Befehl `user` erneut anmelden. Einige Computersysteme, z.B. `ftp.uni-oldenburg.de` erlauben ein anonymes Login mit `user: anonymous` und der email adresse als password (es wird ein `@` erwartet).

Man hat jetzt eine eingeschränkte Kommandozeile zur Verfügung. Es stehen auf dem Remote-Rechner die Befehle `cd`, `dir` und `ls`, `delete`, `mkdir` zur Verfügung.

Die Befehle auf dem eigenen Rechner führt man aus, indem man dem Befehl ein `!` voranstellt, bspw. `!dir`. Hilfe über FTP-Befehle erhält man durch `help`. Eine Verbindung beendet man mit `close`, das Programm mit `bye`.

Das Wesentliche an FTP ist das Austauschen von Dateien zwischen zwei Rechnern. Befindet man sich auf beiden Rechnern im gewünschten Verzeichnis, kann man Dateien von einem Verzeichnis zum anderen senden. Der Befehl `put DATEINAME` sendet eine lokale Datei auf den Remote-Rechner, das Gegenteil macht der Befehl `get DATEINAME`. Will man mehrere Dateien auf einmal versenden, benutzt man `mput`, bspw.

```
binary
prompt
mput *.doc
```

Der Befehl `binary` eins zu eins. Das ist wichtig bei Grafiken, Word-Dokumenten, zip-Dateien und Programmen. Problematisch ist dies bei Textdateien wie C-Quelltexten, HTML oder Latex Sourcen. Diese sollte man im ASCII-Modus (Befehl `ascii`) übertragen. Der Befehl `prompt` schaltet die nervige Aufforderung zur Bestätigung aus, bzw. wieder ein.

## 3.2 Remote arbeiten — telnet

Ein TELNET stellt die Möglichkeit bereit, auf einem Rechner Befehle auszuführen, an dem man gar nicht sitzt, sozusagen Teleworking. Ist also in der Uni der Lieblingsrechner besetzt, dann setzt man sich einfach an einen anderen und arbeitet über das telnet. Dazu startet man das Programm auf der Kommandozeile mit dem Befehl `telnet RECHNERNAME`. Man wird nach Benutzername und Passwort gefragt und bei einigen Systemem nach dem "Terminal". Der entfernte Computer will damit erfahren, welche Zeichen er denn rüberschicken darf und wie diese interpretiert werden. Unter Windows gibt man den Typ `vt100` an, unter Unix/Linux `vt220`. Leider stellen meist nur Unix-Systeme diesen Dienst bereit, d.h. man wird sich in einer Unix-Umgebung wiederfinden. Dort funktionieren die o.g. Kommandos. Anwendungen, die mit einer graphischen Oberfläche laufen(X-Windows), können in dieser Textumgebung natürlich nicht ausgeführt werden<sup>1</sup>.

---

<sup>1</sup>Abhilfe schafft ein sogenannter X-Server und der Export des Displays, hier nicht erklärt.

## 4 Programmieren und Dokumentieren

### 4.1 Die Programmiersprache C

Sehr nah am System kann man mit der Programmiersprache C eigene Programme schreiben, die dann auf der Kommandozeile ausgeführt werden. Diese Programme können sehr schnell laufen und bieten sich deshalb für mathematische Rechnungen und Betriebssysteme an. Eigentlich kann man damit ALLES machen. Wir geben nur ein kleines Programmbeispiel, um das Konzept deutlich zu machen.

```
/* Dies ist ein Beispielprogramm in C*/
/* Kommentare werden in Slash-Sternchen gesetzt */
/* Dateiname: helloworld.c */

/* Einbinden einer Eingabe-Ausgabe Bibliothek: */
#include <stdio.h>

/* Beginn eines Hauptprogrammes mit der Funktion main(),
   der Programmtext wird in geschweifte Klammern gepackt,
   Zeilen werden mit Semikolon abgeschlossen. */
int main(){

    /* Ausgabe einer Zeichenkette mit abschließendem Zeilen-
       umbruch */
    printf("Hello World!\n");

    /* Beendigung des Programms mit dem Integer-Wert 0 */
    return 0;
}
```

Das Programm kann man in jedem beliebigen Texteditor schreiben und verändern. Damit es ausgeführt werden kann, muss der Quellcode kompiliert, also in Maschinensprache übersetzt, werden. Dies macht ein COMPILER. Compiler gibt's von Intel, Microsoft, Borland, GNU usw..., letzterer ist frei verfügbar, funktioniert unter Windows und Unix und läuft in der Kommandozeile. Das Testprogramm `helloworld.c` wird mit dem Befehl `gcc -o helloworld.exe helloworld.c` in ein ausführbares Programm namens `helloworld.exe` übersetzt. Unter Unix sollte man prüfen, ob das Programm Ausführbarkeitsrechte besitzt.

## 4.2 Visualisieren und Programmieren mit MATLAB

Da man mit Cnahe am System programmiert, sind graphische Ausgaben eher mühsam zu programmieren. Auch stellt Cnicht ein fertiges Konzept zum Umgang mit Vektoren und Matrizen oder mathematische Funktionen bereit. Will man sich um das nicht kümmern müssen, dann bietet sich das Visualisierungstool MATLABan. Früher ein "besserer Taschenrechner" für numerische Vektorrechnungen, ist es heute ein mächtiges Programmpaket. Es stellt eine Kommandozeile, eine vollständige Programmiersprache, eine umfangreiche Bibliothek mit mathematischen Funktionen und eine leistungsfähige grafische Ausgabe zur Verfügung.

Ausgeführt wird MATLABüber das "command window", in dem einzelne Befehle eingetippt und sofort ausgeführt werden. Eine Folge von Befehlen kann man in Skripten zusammenfassen und diese als Ganzes aufrufen.

```
% Dies ist ein Kommentar
% Ausgabe des Arbeitsverzeichnisses:
pwd
% Wechseln in temporäres Verzeichnis:
cd 'c:/temp'
% Als kleine Zaubereien vorweg:
peaks(40);
msgbox('hello world!');
% als Beispiel einen Sinus plotten und abspeichern
% x-Vektor definieren:
x=0:0.1:2*pi;
% y-Vektor ausrechnen, hier hilft Matlabs Vektorfähigkeit
y=sin(x);
% Alle definierten Variablen kann man sich anzeigen lassen:
whos
% Funktion in extra Fenster darstellen:
plot(x,y);
% Eigenschaften der Grafik können mit propedit verändert werden:
propedit;
% fertige Graphik als png-Datein abspeichern:
print -dpng 'sinus.png';
```

Kennt man den Namen eines Befehls oder einer Funktion, dann kann man sich mit `help BEFEHLSNAME` eine Hilfe anzeigen lassen. Die Eingabe von `helpwin` oder `helpdesk` öffnet die umfangreiche Hilfe zu MATLAB.

### 4.3 Dokumentieren und wissenschaftlich Textverarbeitung mit $\text{\LaTeX}$

$\text{\LaTeX}$  bietet als Textverarbeitungssystem eine Reihe von Vorzügen. Es zeichnet sich zuallererst durch die Darstellung von mathematischen Formeln aus. Mit  $\text{\LaTeX}$  soll es möglich sein, sich auf die textlichen Inhalte zu konzentrieren, während das Programm ein einheitliches Erscheinungsbild auch großer Dokumente gewährleistet. Eigene Erfahrung zeigt, dass der Hauptvorteil des Systems darin liegt, auch mit mehreren Autoren ohne großen Aufwand ein gemeinsames Dokument zu schaffen.

Formatierungsanweisungen werden im Klartext in das Dokument geschrieben. Dadurch sind die Dateien klein und auf beliebigen Systemen les- und veränderbar. Es bietet sich bei Arbeiten im Netzwerk an. Last but not least ist das Produkt frei.

Eine  $\text{\LaTeX}$ -Quelldatei wird mit dem Befehl `latex DATEINAME` in das .dvi-Format<sup>2</sup>. Zeitgemäßer empfinden wir die Ausgabe als .pdf mit dem Befehl `pdflatex DATEINAME`. PDF-Dokumente bieten hohe Qualität, sind klein und mit dem entsprechenden frei erhältlichen Programm auf jedem System darstellbar.

---

<sup>2</sup>und dieses nachher mit `dvips` in Postscript