

Vorstellung der `tabbing`-Umgebung
sowie der Pakete
`longtable`, `tabularx`, `booktabs` und
`mdwtab`

Stefan Deumlich

Berlin, den 6. Juli 2002

Inhaltsverzeichnis

| | | |
|----------|---------------------------------|-----------|
| 1 | Die tabbing-Umgebung | 2 |
| 2 | Das Paket longtable | 4 |
| 3 | Das Paket tabularx | 7 |
| 3.1 | Die tabular*-Umgebung | 7 |
| 3.2 | Die tabularx-Umgebung | 7 |
| 4 | Das Paket booktabs | 9 |
| 5 | Das Paket mdwtab | 11 |
| 6 | Das Paket hhline | 12 |
| | Literaturverzeichnis | 14 |

Kapitel 1

Die tabbing-Umgebung

Die Umgebung ermöglicht das Setzen von Tabulatoren, wie man sie aus Word kennt. Es wird damit ermöglicht gut strukturierte Dokumente zu erstellen ohne die `tabular`-Umgebung zu benutzen.

Man beachte, dass es in der Umgebung keinen automatischen Zeilenumbruch gibt. Man muss also jede Zeile mit dem Doppelbackslash abschließen und beachten, dass der Text nicht über das Seitenende hinausgeht.

- Tabstopps werden mit `\=` gesetzt und mit `\>` wird zu den Tabs gesprungen
- mit `\kill` wird ein zum Setzen von Tabs genutzte Musterzeile nicht ausgedruckt
- mit `\+` wird der linke Rand des Zeilenbeginns auf den ersten Tab gesetzt
- mit `\-` erreicht man das Gegenteil, aber nicht über den *nullten* Tabstopp hinaus
- `\pushtabs` speichert die aktuellen Tabs und löscht sie dann - ermöglicht Neusetzung
- `\poptabs` stellt die alten Tabs wieder her

Beispiel

```
\begin{tabbing}
  Material\quad\= Qualit"at\quad\= hellbraun\quad\= Preis\kill
  Material \> Qualit"at \> Farbe \> Preis\\[0.5cm]
  Papier   \> mittel    \> wei"s \> niedrig\\
  Leder    \> gut       \> braun \> hoch\\
  Pappe    \> schlecht  \> grau  \> mittel
\end{tabbing}
```

Damit ergibt sich folgender Ausdruck:

| Material | Qualität | Farbe | Preis |
|----------|----------|-----------|---------|
| Papier | mittel | weiß | niedrig |
| Leder | gut | hellbraun | hoch |
| Pappe | schlecht | grau | mittel |

Ein weiteres Beispiel zur Verdeutlichung der Befehle `\pushtabs` und `\poptabs`.

```
\begin{tabbing}
  G"urteltier \= \kill
  M"ucken:\>erschlagen von: \=Menschen\+\+\
    K"uhlen\
    Pferden\-\
    wenig s"attigend\-\
  G"urteltier:\>nicht genie"sbar\
\pushtabs
(siehe auch: \= Ameisenb"ar\
  \> Albatros gesch"utzt)\
\poptabs
Pferde \> gefressen von \> M"ucken
\end{tabbing}
```

ergibt folgendes:

| | | |
|--------------|---------------------|----------|
| Mücken: | erschlagen von: | Menschen |
| | | Kühen |
| | | Pferden |
| | wenig sättigend | |
| Gürteltier: | nicht genießbar | |
| (siehe auch: | Ameisenbär | |
| | Albatros geschützt) | |
| Pferde | gefressen von | Mücken |

Kapitel 2

Das Paket `longtable`

Die Erstellung von Tabellen ist mit der `tabular`-Standardumgebung auf Tabellen beschränkt, deren Länge die vorgegebene Seitenhöhe nicht überschreitet. Sollen längere Tabellen erstellt werden, so müsste man die Tabelle auf mehrere `tabular`-Umgebungen aufteilen. Dies verursacht aber weitere Probleme, so werden die Tabellen unterschiedlich breit sein, da \LaTeX die Größe der Tabellen automatisch an den längsten Spalteneintrag anpasst.

Das Paket `longtable` von D. P. CARLISLE löst diese Schwächen. Der Befehl wird wie folgt genutzt:

```
\begin{longtable}{sp_form}  Zeilen  \end{longtable}
```

Die Spalten *sp_form* werden ganz normal, wie in der `tabular`-Umgebung angegeben (vgl. [1] S. 26-27). Weiterhin stehen folgende neue Befehle zur Verfügung:

- `\endhead` - Ende der Zeile, die am Kopf einer Tabelle auf jeder Seite erscheint
- `\endfirsthead` - Ende der Kopfzeile der ersten Seite
- `\endfoot` - Ende der Tabellenfußzeile auf jeder Seite
- `\endlastfoot` - Ende der Tabellenfußzeile auf der letzten Seite

Das heißt, dass man auf der ersten Seite einen anderen Titel oberhalb der Tabelle hat als auf allen anderen Seiten. Ebenso verhält es sich mit dem Tabellenfuß. `\endlastfoot` ermöglicht am Tabellenende einen anderen Text als auf allen anderen Seiten.

Weiterhin hat man die Möglichkeit mit `\caption{text}` der Tabelle einen Titel zu geben, der dann auch im Tabellenverzeichnis erscheint. Die Tabelle erhält eine Nummer, die die Kapitelnummer als erste Zahl hat und dann für jede Tabelle eine weitere Zahl hinzugefügt wird. Z.B. Tabelle 2.1 oder 2.2 usw. Daneben gibt es auch die *-Form des `\caption`-Befehls. Damit erscheint diese Tabelle nicht im Tabellenverzeichnis und es wird ebenfalls die Anzeige der Tabellennummer über der Tabelle unterdrückt.

Betrachten wir uns nun ein Beispiel, zuerst erscheint der Quelltext gekürzt. Alle angesprochenen Befehle sind jedoch zu erkennen. Danach folgt die dem Quelltext entsprechende Tabelle:

```

\begin{longtable}{|lr|}
  \caption{\bfseries Naturparke in Deutschland}\hline
  Bezeichnung & Fläche & $(\mbox{km}^2)$\hline
  \endfirsthead
  \caption*{\bfseries Naturparke -- Fortsetzung}\hline
  Bezeichnung & $(\mbox{km}^2)$\hline
  \endhead
  \hline\multicolumn{2}{|r|}{\textit{Fortsetzung auf der
    nächsten Seite}}\hline
  \endfoot
  \hline\multicolumn{2}{|l|}
  {\textit{Quelle}: Verband Deutscher Naturparke, 1987}\hline
  \endlastfoot
  \slshape Baden-W"urtemberg: & \\\
  Neckar-Odenwald & 1300\\
  Obere Donau & 825\\
  ...
  \slshape Saarland: & \\\
  Saar-Hunsr"uck & 1662
\end{longtable}

```

Tabelle 2.1: **Naturparke in Deutschland**

| Bezeichnung | Fläche (km ²) |
|---|------------------------------|
| <i>Baden-Württemberg:</i> | |
| Neckar-Odenwald | 1300 |
| Obere Donau | 825 |
| Schönbuch | 155 |
| Schwäb.-Fränk. Wald | 900 |
| Stromberg-Heuchelberg | 328 |
| <i>Bayern:</i> | |
| Altmühltal | 2908 |
| Bayrische Rhön | 1240 |
| Bayrischer Spessart | 1710 |
| Bayrischer Wald | 2068 |
| Fichtelgebirge | 1004 |
| Fränkische Schweiz | 2348 |
| Frankenhöhe | 1070 |
| Frankenwald | 1116 |
| Haßberge | 823 |
| <i>Fortsetzung auf der nächsten Seite</i> | |

Naturparke – Fortsetzung

| Bezeichnung | (km ²) |
|---------------------------|--------------------|
| Hessenreuther Wald | 270 |
| Steigerwald | 1280 |
| Steinwald | 232 |
| <i>Hamburg:</i> | |
| Harburger Berge | 38 |
| <i>Hessen:</i> | |
| Bergstraße-Odenwald | 1603 |
| Diemelsee | 334 |
| Habichtswald | 471 |
| Hessische Rhön | 700 |
| Hessischer Spessart | 710 |
| Hochtaunus | 1201 |
| Hoher Vogelsberg | 384 |
| Meißner-Kaufunger-Wald | 429 |
| Rhein-Taunus | 807 |
| <i>Schleswig-Holstein</i> | |
| Aukrug | 380 |
| Hüttener Berge Wittensee | 260 |
| Holsteinische Schweiz | 523 |
| Lauenburgische Seen | 444 |
| Westensee | 260 |
| <i>Saarland</i> | |
| Saar-Hunsrück | 1672 |

Quelle: Verband Deutscher Naturparke, 1987

Kapitel 3

Das Paket `tabularx`

3.1 Die `tabular*`-Umgebung

L^AT_EX erlaubt mit der `*`-Form der `tabular`-Umgebung die Vorgabe der Breite einer Tabelle.

Syntax

```
\begin{tabular*}{breite}[pos]{sp_form}
```

Zeilen

```
\end{tabular*}
```

Innerhalb der Umgebung werden die Spaltenbreiten automatisch ermittelt oder mit `p{sp_breite}` angegeben. Die automatische Breitenenerkennung orientiert sich immer am längsten Eintrag einer Spalte. Mit der vordefinierten Spaltenbreite wird der Text in den Spalten automatisch umbrochen und man benötigt keinen manuellen Zeilenumbruch. Dies ist von Vorteil, wenn man lange Texte in Tabellen bringen will. Dies würde mit der normalen `tabular`-Umgebung zu unschönen Ergebnissen führen.

3.2 Die `tabularx`-Umgebung

Um dieses Paket zu nutzen, muss man dieses extra mit dem Befehl `\usepackage{tabularx}` in der Dokumentenpräambel einbinden.

Die `tabularx`-Umgebung hat dieselbe Syntax wie `tabular*`, zusätzlich jedoch den Spaltenformatierungsparameter `X`. Die Breite der `X`-Spalten berechnet sich automatisch aus der Gesamtbreite der Tabelle. Andere Spaltenangaben gibt man wie gewohnt an. Ein weiterer Vorteil der `X`-Spalten ist, dass in ihnen Fußnoten erlaubt sind. Somit wird ein weiteres Manko von den Standardumgebung behoben.

Beispiel

```
\begin{tabularx}{120mm}{lXcXr}
```

Im Beispiel wird eine 120mm breite Tabelle definiert, die fünf Spalten hat. Die erste Spalte ist linksbündig, die dritte zentriert und die fünfte rechtsbündig. Die Breite dieser Spalten ergibt sich jeweils aus dem längsten Eintrag. Die zweite und vierte Spalte sind X-Spalten. Die Breite dieser Spalten ergibt sich erst nachdem L^AT_EX festgestellt hat, wie breit die anderen drei Spalten sind und berechnet dann die Breite der beiden Spalten. Sind die drei Spalten zum Beispiel zusammen 60mm breit, dann werden die beiden X-Spalten jeweils 30mm breit sein. Deshalb sollte man auch vorher überlegen, wie breit die anderen Spalten werden. Sonst kann es passieren, dass die X-Spalten zu klein werden.

Es gibt auch noch weitere Möglichkeiten die X-Spalten entsprechend seinen Wünschen anzupassen. Eine etwas umständliche Form ist die direkte Eingabe in der Tabellendeklaration:

```
>{\Befehle}sp_form
```

Etwas eleganter geht dies mit den X-Spalten. Dabei definiert man sich einfach einen neuen Spaltentyp mit

```
\newcolumnntype{Y}{>{\Befehle}X}
```

Möchte man den von L^AT_EX voreingestellten Blocksatz in den Spalten abschalten, so kann man dies mit dem Befehl `\raggedright` tun, dieser erzeugt Flattersatz. Allerdings führt das zu einem Nachteil, denn dieser Befehl definiert den Zeilenumbruch „\“ neu. Dies kann man jedoch umgehen, in dem man den Befehl immer zusammen mit `\arraybackslash` benutzt. Somit kann man sich also eigene Spaltentypen mit den X-Spalten definieren. Betrachten wir uns dazu ein weiteres Beispiel:

```
\newcolumnntype{Y}{>{\small\raggedright\arraybackslash}X}  
\begin{tabularx}{\linewidth}{|Y|Y|Y|}
```

...

Kapitel 4

Das Paket booktabs

Betrachten wir uns zunächst ein einführendes Beispiel. Die erste Tabelle benutzt das Paket `booktabs`, die zweite Tabelle versucht die erste Tabelle mit den normalen \LaTeX -Mitteln nachzuempfinden.

| Item | | |
|-----------|-------------|------------|
| Animal | Description | Price (\$) |
| Gnat | per gram | 13.65 |
| | each | 0.01 |
| Gnu | stuffed | 92.50 |
| Emu | stuffed | 33.33 |
| Armadillo | frozen | 8.99 |

normales \LaTeX

| Item | | |
|-----------|-------------|------------|
| Animal | Description | Price (\$) |
| Gnat | per gram | 13.65 |
| | each | 0.01 |
| Gnu | stuffed | 92.50 |
| Emu | stuffed | 33.33 |
| Armadillo | frozen | 8.99 |

Man erkennt deutlich die Unterschiede zwischen beiden Tabellen. So sieht man, dass die Abstände zwischen Text und Linien in der unteren Tabelle viel kleiner sind als in der oberen. Auch sind die Linien im oberen Teil unterschiedlich dick, was in der unteren Tabelle nicht der Fall ist.

Um das `booktabs`-Paket zu nutzen, muss es wieder in der Dokumentenpräambel mittels `\usepackage{booktabs}` eingebunden werden. Mit dem Paket kann man die `tabular`-Umgebung ganz normal wie sonst auch benutzt werden. Es werden aber einige zusätzliche

Befehle bereitgestellt. Diese beziehen sich hauptsächlich auf die Trennlinien zwischen Zeilen. Sie bieten eine verbesserte Abstandsverarbeitung:

- mit `\toprule[breite]` beginnt eine Tabelle
- mit `\bottomrule[breite]` endet sie entsprechend
- in der Tabelle benutzt man `\midrule[breite]`
- für einzelne Zellen `\cmidrule` mit der Syntax `\cmidrule[breite](trim){a-b}`
- mit `\addlinespace[breite]` erzeugt man einen Extraabstand zwischen Zeilen

Betrachten wir uns nun noch einmal das zu Beginn vorgestellte Beispiel. Zuerst der Quelltext:

```
\begin{tabular}{@{}llr@{}} \toprule
\multicolumn{2}{c}{Item}\cmidrule(r){1-2}
Animal & Description & Price (\$)\cmidrule
Gnat   & per gram    & 13.65\\
       & each        & 0.01\\
Gnu    & stuffed     & 92.50\\
Emu    & stuffed     & 33.33\\
Armadillo & frozen     & 8.99\\ \bottomrule
\end{tabular}
```

Dies führt zu dem bereits bekannten Ergebnis:

| Item | | |
|-----------|-------------|------------|
| Animal | Description | Price (\$) |
| Gnat | per gram | 13.65 |
| | each | 0.01 |
| Gnu | stuffed | 92.50 |
| Emu | stuffed | 33.33 |
| Armadillo | frozen | 8.99 |

Kapitel 5

Das Paket `mdwtab`

Dieses Paket wird in der anderen Datei vorgestellt, da das `mdwtab`-Paket die kompletten Tabellenbefehle neu definiert. Dies hätte hier dazu geführt, dass man einige Unterschiede nicht hätte darstellen können.

Kapitel 6

Das Paket `hhline`

Betrachten wir uns wieder zu Beginn ein Beispiel:

| | | | |
|---|---|---|---|
| a | b | c | d |
| 1 | 2 | 3 | 4 |
| i | j | k | l |
| w | x | y | z |

Dieses Beispiel sieht nicht sehr schön aus. Die Ecken der Tabelle werden nicht sehr ansprechend umgesetzt. Außerdem hätten wir gerne auch innerhalb der Tabelle doppelte Linien. Dies lässt sich aber mit den normalen \LaTeX -Mitteln nicht umsetzen.

Doch es gibt eine Lösung für unser Problem. Der uns schon weiter oben bekannte DAVID CARLISLE hat das `hhline`-Paket entwickelt, das diese Missstände beseitigt. Wie üblich müssen wir das Paket erst mittels `\usepackage{hhline}` in der Dokumentenpräambel einbinden. Das Paket stellt uns den Befehl `\hhline` zur Verfügung, der ähnlich `\hline` arbeitet, aber ein verbessertes Zusammenspiel mit vertikalen Linien bietet. `\hhline` wird eine Parameterliste übergeben. Folgende Parameter stehen zur Verfügung:

- = erzeugt eine horizontale Doppellinie von der Breite der zugehörigen Spalte.
- erzeugt eine horizontale Einfachlinie von der Breite der zugehörigen Spalte.
- ~ unterdrückt die horizontale Linienstruktur für die zugehörige Spalte.
- | erzeugt eine vertikale Linie zwischen zwei Spalten von der Höhe der horizontalen Doppellinie.
- : erzeugt eine *unsichtbare* vertikale Linie zwischen zwei Spalten von der Höhe einer horizontalen Doppellinie, an die sich die nächste horizontale Linie anschließt oder an der die vorangehende horizontale Linie endet.
- # erzeugt ein horizontales Doppelliniensegment von der Breite einer vertikalen Doppellinie.

`t` erzeugt die obere Linie eines horizontalen Doppelliniensegments von der Breite einer vertikalen Doppellinie.

`b` wie `t`, jedoch für die untere Linie des horizontalen Doppelliniensegments.

(Vgl. dazu auch [1] S. 24)

Betrachten wir nun noch einmal das einführende Beispiel, diesmal aber wie wir uns es vorgestellt haben. Zuerst der Quelltext:

```
\begin{tabular}{||cc||c|c||}\hhline{|t==:t==:t|}
  a & b & c & d \\\hhline{!:=~|~|}
  1 & 2 & 3 & 4 \\\hhline{#=#~|=#}
  i & j & k & l \\\hhline{||--||--||}
  w & x & y & z \\\hhline{|b==:b==:b|}
\end{tabular}
```

Führt zu folgendem Ergebnis:

| | | | |
|---|---|---|---|
| a | b | c | d |
| 1 | 2 | 3 | 4 |
| i | j | k | l |
| w | x | y | z |

Literaturverzeichnis

- [1] KOPKA, H., 1997: \LaTeX Ergänzungen - Band 2, Bonn, 2., überarb. Auflage, Addison Wesley
- [2] KOPKA, H., 1996: \LaTeX Einführung - Band 1, Bonn, 2., überarb. Auflage, Addison Wesley
- [3] GOOSSENS, M., MITTELBACH, F., SAMARIN, A., 2000: Der \LaTeX -Begleiter, Paris, Addison Wesley
- [4] Dokumentation zum `booktabs`-Paket; zu finden unter `docs/booktabs`
- [5] Dokumentation zum `mdwtab`-Paket; zu finden unter `docs/mdwtools`