

7 Tables Continued

7.1 Longtable

The `tabular` and `tabularx` environments explained in the previous chapter provide a convenient way of making tables. These environments, however, fail if a table exceeds one page. One easy trick that might do the job would be to break the tables manually but the `longtable` package enables automatic page breaks by the \TeX compiler.

7.1.1 Constructing longtables

The `longtable` environment shares most of the features with the `tabular` environment. We begin with the following example that uses most of the features of the `longtable` environment.

Table 7.1: A long table

This part appears at the top of the table				
FIRST ¹	SECOND	*	THIRD	
This table is only <i>slightly</i> different from the guide to the <code>longtable</code> Columns 1 & 2 here have fixed widths. <code>2.5in</code> <code>longtable</code> columns are specified same way as in the <code>tabular</code> <code> p{2.5in} p{1in}@{*}c </code> Each line ends with a <code>\</code> command. The <code>\</code> command has an argument, just as <code>tabular</code> See the effect of <code>\\[10pt]</code> below: The <code>\tabularnewline</code> command is an alternative to <code>\</code> for use in the scope of <code>\raggedleft</code> and similar commands that redefine <code>\</code> . Notice <code>\tabularnewline[10pt]</code> below: Some lines can be really long: This column is a “p” column so that this row of the table can take up several lines. But \TeX will never break a page within such a row. <code>*</code> has the same effect as <code>\</code> disallows a page break after <code>setlongtables</code> is an obsolete v4.09 of <code>longtable</code> and does This goes at the	one in the package.	*	1	
			*	2
			*	3
		<code>1in</code>	*	
		in the environment.	*	5
			*	6
		in this case.	*	7
		<code>\</code> command.	*	8
		optional	*	9
		in the environment.	*	10
			*	11
			*	12
			*	13
			*	14
	This is also a “p” column.	*	This is not.	
	but it	*	16	
	the row.	*	17	
	command in	*	18	
	nothing.	*	19	
	bottom.	*	0.00	

¹ You can also have a footnote in the table head by using `\footnotemark` and `\footnotetext`.

Table 7.1: (continued)

This part appears at the top of every other page			
	First	Second	Third
Center aligned text		◇	* 20
Right aligned text		○	* 21
Also <code>\hline</code> can be used as in <code>tabular</code> .			* 22
That was a <code>\hline</code> .			* 23
That was a <code>\hline\hline</code> .			* 24
This line is produced by <code>\multicolumn²...</code>			
Lots of lines		like this.	* 25
Lots of lines		like this.	* 26
Lots of lines		like this.	* 27
Lots of lines		like this.	* 28
Lots of lines		like this.	* 29
	One ³ line	like this.	* 30
Another one		like this ⁴	* 31
Lots of lines		like this.	* 32
	These lines will	appear	* 1.00
	in place of the	usual foot	* 2.00
	at the end	of the table.	* 3.00

7.1.2 Optional arguments

The optional arguments to `\begin{longtable}` are:

- c** The table is set center aligned.
- l** The table is set flush left.
- r** The table is set flush right.

If no arguments are specified, the position of the table is set according to the values of `\LTleft` and `\LTright`.

7.1.3 Commands and parameters

This section contains a brief description of the commands and other parameters that may be used in the `longtable` environment.

LTchunksize This corresponds to the number of rows that \TeX has to keep in memory at one time. By default this value is set to 20, but it can be set by the user; for instance, by `\setcounter{LTchunksize}{10}` or `\LTchunksize=10` to a value of 10. Changing the default does not affect page breaking. However, \TeX will run faster with a large `LTchunksize`, and, on the other hand, will require more memory. The minimum value of `LTchunksize` can be set equal to 1; however, it must be at least as large as the number of rows in each of the head or foot sections (if the table head and foot need to be set).

\LTleft The defaults in the `longtable` package are such that the tables are set flush left, but are indented by the usual paragraph indentation. `\LTleft` controls the amount of glue to the left of the table. By default this is set to `\parindent`, but can be changed according to the requirements; for instance, by `\setlength\LTleft{0pt}`.

² We had seen more use of `\multicolumn` in the previous chapter.

³ This is another footnote in the body (obtained by `\footnote`).

⁴ Yet another footnote.

<code>\LTright</code>	This parameter determines the glue to the right of the table. The default for this is <code>\fill</code> .
<code>\LTpre</code>	Denotes the glue before the table. The default is set to <code>\bigskipamount</code> .
<code>\LTpost</code>	Denotes the glue after the table. The default is again <code>\bigskipamount</code> .
<code>\LTcapwidth</code>	This controls the width of the parbox containing the caption. The default width is set to <code>4in</code> , but can be changed, for instance, to a value of <code>2in</code> by using <code>\setlength\LTcapwidth{2in}</code> .
<code>\endhead</code>	At the start of the table, one may specify the lines that are to appear at the top of every page. This command is used in place of the last <code>\</code> .
<code>\endfirsthead</code>	If the head on the first page needs to be different than on the others, then one may specify the lines to appear in a normal way followed by <code>\endfirsthead</code> (in place of the last <code>\</code>).
<code>\endfoot</code>	Specifies rows to appear at the bottom of every page.
<code>\endfirstfoot</code>	Specifies rows to appear at the bottom of the last page. The commands <code>\endfirsthead</code> and <code>\endlastfoot</code> are useful when one wants to specify something that should logically appear in the table at the end of the <code>firsthead</code> , or at the beginning of the <code>lastfoot</code> .
<code>\</code>	This is similar to the <code>tabular</code> environment and specifies the end of the row.
<code>\</code> [<i>dim</i>]	This is also similar as in the <code>tabular</code> environment and marks the end of the row, and then adds vertical space (as shown in Table (7.1)).
<code>\</code>*	It is the same as <code>\</code> , but disallows a page break after the row.
<code>\tabularnewline</code>	It is an alternative to <code>\</code> for use in the scope of <code>\raggedright</code> and similar commands that redefine <code>\</code> . It can also be used with an optional argument, <code>\tabularnewline</code> [<i>dim</i>], so as to specify the end of the row, and then add a vertical space.
<code>\kill</code>	If a line is <code>\killed</code> , by using <code>\kill</code> rather than <code>\</code> at the end of the line, it is used in calculating column widths, but removed from the final table.
<code>\pagebreak</code>	Forces a page break.
<code>\pagebreak</code> [<i>val</i>]	A ‘hint’ between 0 and 4 of the desirability of a page break. A high value indicates more desirability.
<code>\nopagebreak</code>	Prohibits a page break.
<code>\nopagebreak</code> [<i>val</i>]	A ‘hint’ between 0 and 4 of the undesirability of a page break.
<code>\newpage</code>	Forces a page break.
<code>\caption</code>{<i>caption</i>}	Caption ‘Table ?: <i>caption</i> ’, and a ‘ <i>caption</i> ’ entry in the list of tables.
<code>\caption</code> [<i>lot</i>]{<i>caption</i>}	Caption ‘Table ?: <i>caption</i> ’, and a ‘ <i>lot</i> ’ entry in the list of tables.
<code>\caption</code> [] {<i>caption</i>}	Caption ‘Table ?: <i>caption</i> ’, but no entry in the list of tables.
<code>\caption</code>*{<i>caption</i>}	Caption ‘ <i>caption</i> ’, but no entry in the list of tables.
<code>\footnote</code>	Used for having footnotes, but it cannot be used in the table head and foot.
<code>\footnotemark</code>	Footnotemark, can be used only in the table head and foot.
<code>\footnotetext</code>	Footnote text: for use in the table body after a <code>\footnotemark</code> has been set (should appear on the page on which the footnote is desired).

Table 7.2: (continued)

Another long table example (continued)		
First two columns		Third column
Lots of line like	this	2
Lots of line like	this	2
Lots of line like	this	2
Lots of line like	this	2
Lots of line like	this	2
The End		

7.2.1 Verbatim highlights from Table (7.2):

```

\begin{longtable}[p{0.9in}|p{2in}|c|]
xxxxx & xxxxx & xxxxxxxxxxxxxxxx \kill
\caption{A simple example\label{simple}}\ \hline\hline
\multicolumn{3}{c}{\bf Another long table example}\ \hline\hline
\multicolumn{2}{c}{First two columns} & {Third column}\ \hline
\multicolumn{2}{c}{p-type} & \ \hline\hline
\endfirsthead
\caption[] {(continued)}\ \hline\hline
\multicolumn{3}{c}{\bf Another long table example (continued)}\ \hline\hline
\multicolumn{2}{c}{First two columns} & {Third column}\ \hline
\endhead
\hline
\multicolumn{3}{c}{\bf Continued $\ldots$}\ \hline
\endfoot
\hline
\multicolumn{3}{c}{\bf The End}\ \hline
\endlastfoot
p column & another one & 1 \ \hline
Lots of line like & this & 2 \ \hline
...
\end{longtable}

```

7.3 Exercise

Try making Table (7.1); it seems to be quite strange, but it contains most of the features one would need in order to get thoroughly acquainted with the `longtable` environment. It should be *quite easy* if one actually reads the table itself. Look at the spaces *carefully*.