The update of nostarch class

Boris Veytsman\textsuperscript{1}

\textsuperscript{1}TUG, Chan Zuckerberg Initiative, George Mason University
No Starch Press publishes the finest in geek entertainment—bestsellers like Python Crash Course, Python for Kids, How Linux Works, and Hacking: The Art of Exploitation. We focus on computer programming, security, hacking, alternative operating systems, STEM, and LEGO. Our titles have personality, our authors are passionate, and we read and edit everything we publish.
Some titles

https://nostarch.com/catalog.htm
History:

1. Many distinctive typographic features → must accommodate them.
2. The previous version was released in 2008. A lot of in house changes by No Starch team. Proliferation of small hacks.
3. 2023: the need to create a new unified class.

Design principles:

1. Automate as much a possible.
2. Leave the last 5% for manual tweaking.

This talk: TeXnical problems and their solutions.

No Starch solution: always split before hyphen: https://long-url.com!
URL splitting


No Starch solution: always split before hyphen: https://long-url.com!

The rules: No Starch house style

1. Split after # > ] ) } : 
2. Split before . _ = & - ! ? | , ; @ ' " + 
3. Never split before / (including https://)

No Starch solution: always split before hyphen: https://long-url.com!

The rules: No Starch house style

1. Split after # > ] ) } :
2. Split before . _ = & - ! ? | , ; @ ’ " +
3. Never split before / (including https://)

Note: In URLs the symbols ? ! & often start a group, as in https://www.google.com/search?q=tex
URL splitting implementation

Using *url.sty* by Donald Arseneau. An URL is internally a *math* expression.

Splits after a symbol: declare the symbol a mathrel or mathbin!

How can we make a split *before* a symbol?

Solution: *UrlSpecials*. Here we can do what we want!

Example 1: Break before dot:

\g@addto@macro\{\UrlSpecials\}{%
\do\.{\penalty\UrlBreakPenalty\mathchar46\relax}%
\}

Example 2: No break before slash (but can break after, unless followed by slash):

\g@addto@macro\{\UrlSpecials\}{%
\do\/\{\unpenalty\penalty@M\mathchar47\penalty\UrlBreakPenalty\}%
\}
URL splitting: unsolved problem

*amsmath* redefines open delimiters in math mode ⇒ a strange error message

\g@addto@macro{\UrlSpecials}{%  
  \do\{{\penalty\UrlBreakPenalty\mathchar40\relax}}

is fine without *amsmath*. With *amsmath*:

Bad mathchar (32768)

Redefenition of \@afterheading:

1. Add the parshape and larger font to \everypar.
2. At the end of \everypar redefine everypar.

Manual override: \turnoffbigpara disables this behavior.
Figures and tables are not centered, and the width of the caption is the same as the width of the figure or table.
Captions implementation: setting in a parbox

\DeclareCaptionFormat*{nostarchfigtblformat}{% 
  \parbox{\nostarch@captionwidth}{#1#2#3}}

We need to calculate \nostarch@captionwidth.
Captions implementation: figures

At the start of the caption: look at the width of the last box, and set the caption width to it!

We add to figure caption code (using `etoolbox`)

\nostarch@measurecaptionwidth:

\newcommand\nostarch@measurecaptionwidth{% 
\ifnostarch@overridecaptionwidth\else 
  \par 
  \setbox\@tempboxa\lastbox 
  \setbox\@tempboxa=\hbox{\unhbox\@tempboxa}\
  \global\setlength{\nostarch@captionwidth}{\wd\@tempboxa}\
  \par 
\fi
\global\nostarch@overridecaptionwidthfalse}

Boris Veytsman

The update of nostarch class

TUG2023 12 / 18
Captions implementation: tables. I. Algorithm

Problem: table captions are before tables.

Solution: save the widths to the .aux file, and read them.

The ideas are mostly stolen from longtable package.

1. For table number $N$ define a macro \texttt{nostarch@tbl@Roman\{N\}}.
2. At the end of the table write its width to the macro and save to the .aux file
3. If the macro exists, check whether the width changed to trigger rerun.
4. At the beginning of table check if the macro exists, and if yes, typeset the caption to this width.
Boris Veytsman

The update of nostarch class

TUG2023 14/18

Captions implementation: tables. II. Writing

\nostarch@measurecaptionwidth
\if@filesw
  \immediate\write@auxout{%
    \gdef\expandafter\noexpand\csnamenostarch@tbl@\romannumeral\c@nostarch@tbl\endcsname
    {\the
      \nostarch@captionwidth}}%
\fi
\expandafter\ifx\csname nostarch@tbl@\romannumeral\c@nostarch@tbl\endcsname\relax
  \edef\@tempa{\the\textwidth}%
\else
  \edef\@tempa{\csname nostarch@tbl@\romannumeral\c@nostarch@tbl\endcsname}%
\fi
\expandafter\ifdim\@tempa=\nostarch@captionwidth\else
  \gdef\nostarch@tbl@warn{\ClassWarningNoLine{nostarch}
    {Table widths have changed. Rerun LaTeX.}}
\fi
\newcommand\nostarch@read@tbl@width{%  
\stepcounter{nostarch@tbl}%
\expandafter\ifx\csname nostarch@tbl@\romannumeral  
c@nostarch@tbl\endcsname\relax
  \edef\@tempa{\the\textwidth}%
\else
  \edef\@tempa{\csname nostarch@tbl@\romannumeral  
c@nostarch@tbl\endcsname}%
\fi
\expandafter\setlength\expandafter\nostarch@captionwidth
\expandafter{\@tempa}%
}
Captions implementation: longtables

Longtables know the width of each column. They are stored in \LT@Roman\{N_{LT}\}. We just add them up.

\def\nostarch@read@LT@width{%
  \ifnostarch@overridecaptionwidth\else
    \global\setlength\{\nostarch@captionwidth\}\{\z@\}%
    \bgroup
      \def\LT@entry##1##2{%
        \global\addtolength\{\nostarch@captionwidth\}\{##2\}%
        \csname LT@\romannumeral\c@LT@tables\endcsname
      }%egroup
  \fi
  \global\nostarch@overridecaptionwidthfalse
}
Captions implementation: override

The user is able to override the automatic setting:

\NextCaptionWidth{2in}

Implementation is simple:

\newcommand\NextCaptionWidth[1][1]{%
  \global\nostarch@overridecaptionwidthtrue
  \global\setlength{\nostarch@captionwidth}{#1}%
}
1. No Starch Press provides unusual challenges, which can be met with some TeX tricks!
2. Next steps: ebooks!