fancyhdr und scrlayer in trauter Zweisamkeit

Abstract
Auf CTAN gibt es diverse Pakete, mit denen man den Seitenstil eines Dokuments verändern kann. Eines der beliebtesten dürfte dabei fancyhdr sein. Ein anderes, eher grundlegendes Paket ist das KOMA-Script-Paket scrlayer. Dieses Paket geht mit seinem Ebenenmodell weit über das hinaus, was ein Seitenstilpaket üblicherweise leistet. Daher wurde schon bald nach Veröffentlichung von scrlayer der Wunsch geäußert, die Ebenen von scrlayer zusammen mit den Seitenstilen von fancyhdr verwenden zu können. Mit dem experimentellen Paket scrlayer-fancyhdr ist dies nun möglich.

Was ist fancyhdr?

Das Paket fancyhdr stellt im Wesentlichen den neuen Seitenstil fancy bereit. Dieser besteht in Kopf und Fuß aus jeweils einem Element, das linksbündig gesetzt wird, einem Element, das zentriert gesetzt wird, und einem Element, das rechtsbündig gesetzt wird. Im doppelseitigen Satz sind auf linken Seiten das linksbündige und das rechtsbündige Element vertauscht. Diese sechs Elemente des Seitenstils können über die Anweisungen \fancyhead und \fancyfoot oder alternativ mit \lhead, \chead, \rhead, \lfoot, \cfoot und \rfoot mit Inhalt versehen werden. Darüber hinaus, gibt es Befehle, um die Dicke einer Linie unter dem Kopf und über dem Fuß dieses Seitenstils einzustellen.

Zusätzlich gibt es auch noch eine Anweisung \fancypagestyle, über die man weitere Seitenstile definieren kann. Intern handelt es sich allerdings bei all diesen zusätzlichen Stilen ebenfalls um fancy. Bei der Aktivierung der neuen Seitenstile werden lediglich zunächst die bei der Definition mit \fancypagestyle als zweites Argument angegebenen Anweisungen ausgeführt.

Was die meisten Anwender nicht wissen ist, dass fancy intern wiederum auf dem Seitenstil @fancy basiert. Außerdem stellt fancyhdr auch noch einen nicht dokumentierten Seitenstil fancyplain bereit. Dieser aktiviert nicht nur den Stil fancy, sondern ändert zusätzlich den Standard-Seitenstil plain in den ebenfalls internen Seitenstil plain@fancy, der auch auf @fancy basiert. Verwendet man diesen nicht dokumentierten Seitenstil fancyplain, so kann man Unterschiede zwischen den Seitenstilen fancy und plain bei den Einstellungen für fancy über die Anweisung \fancyplain festlegen. In der Voreinstellung verwendet fancyhdr beispielsweise

```
\fancyhead[l]{\fancyplain{}{\slshape\rightmark}}
```

um im linken Element des Kopfes für den plain-Seitenstil (erstes Argument von \fancyplain) keinen Kolumnentitel zu setzen, während für fancy (zweites Argument von \fancyplain) die rechte Marke in schräger Schrift ausgegeben wird.

Anzumerken wäre außerdem noch, dass der Seitenstil @fancy und damit alle von fancyhdr definierten und verwendeten Seitenstile genau wie headings automatische Kolumnentitel aktiviert. Die Verwendung eines mit fancyhdr definierten Seitenstils mit rein manuellen Kolumnentiteln ist daher nur möglich, wenn man in der Definition der Seitenstile direkt den gewünschten Kolumnentitel angibt, statt mit

\leftmark und \rightmark Platzhalter zu setzen und diese im Dokument dann mit \markboth und \markright zu befüllen. Erfahrene Anwender wissen, dass dieses direkte Umdefinieren des Seitenstils in einigen Fällen zu falschen Kolumnentiteln führen kann.

Zusammenfassend kann man also sagen, dass fancyhdr es dem Anwender ermöglicht den Inhalt von Kopf und Fuß der Seite über jeweils drei Elemente einzustellen. Die meisten Anwender schätzen die einfach gehaltene Schnittstelle und kommen mit den dokumentierten Möglichkeiten zur Definition eines Seitenkopfes und Seitenfußes sehr gut aus. Näheres zu den dokumentierten Möglichkeiten des Pakets ist [6] zu entnehmen.

Was ist scrlayer?

Das Paket scrlayer geht weit darüber hinaus nur den Kopf und Fuß einer Seite konfigurierbar zu machen. Es führt dazu ein Ebenenmodell ein, bei dem im Hintergrund und im Vordergrund der Seite eine ganze Reihe an Darstellungsebenen übereinander gestapelt werden können. Ebenen können sogar mehrfach verwendet werden. Seitenstile dienen dabei einerseits als Anker für die Ebenen, andererseits werden Seitenstile durch diese Ebenen selbst definiert. Auf diesem Weg ist es über einen Seitenstil nicht nur möglich, den Inhalt des Kopfes und des Fußes zu bestimmen. Es kann auch Material an beliebigen Stellen einer Seite platziert werden.

Eine neue Ebene wird über die Anweisung \DeclareNewLayer deklariert. Bei dieser Deklaration können über Optionen eine ganze Reihe von Eigenschaften der Ebene bestimmt werden. Der Inhalt der Ebene wird beispielsweise als Wert von contents angegeben. Über weitere Optionen kann die Position dieses Inhalts auf der Seite, die Ausgabe der Ebene im Hinter- oder Vordergrund der Seite und vieles mehr eingestellt werden. Derartige Ebenen können dann über \DeclarePageStyleByLayers zu einem Seitenstil kombiniert werden. Es ist auch möglich, Ebenen nachträglich zu ändern oder zu Seitenstilen hinzuzufügen oder von diesen zu entfernen. Grundvoraussetzung ist, dass der Seitenstil irgendwann per \DeclarePageStyleByLayers definiert wurde.

Das Paket scrlayer erhebt auf der anderen Seite nicht den Anspruch, eine einfache Benutzerschnittstelle zur Bestimmung des Inhalts von Kopf und Fuß der Seite bereit zu stellen. Stattdessen überlässt es diese Aufgabe spezialisierten Schnittstellenpaketen. Mit scrlayer-scrpage existiert in KOMA-Script beispielsweise so ein Paket, das unter anderem ähnlich fancyhdr Seitenstile mit dreigeteiltem Kopf und Fuß bereitstellt. Näheres zu scrlayer-scrpage ist [4] zu entnehmen. Darüber hinaus wurden in »Die TEXnische Komödie« 3/2017 in den Artikeln [1], [2] und [3] weitere Anwendungsmöglichkeiten für scrlayer beispielhaft vorgeführt.

Wie mächtig die Möglichkeiten von scrlayer sind, zeigt das ebenfalls in [4] dokumentierte Schnittstellenpaket scrlayer-notecolumn, das basierend auf scrlayer die Möglichkeit schafft, beliebige Notizspalten auf Seiten zu platzieren, wobei die Notizen auch automatisch über mehrere Seiten umbrochen werden können. Die Pakete scrlayer-scrpage und scrlayer-notecolumn stehen dabei nicht in Konkurrenz zueinander, sondern können auch zusammen verwendet werden.

Zusammenfassend kann man also sagen, dass scrlayer LaTEX mit Hilfe von Seitenstilen sehr grundlegend um die Fähigkeit von Ebenen erweitert. Die Bereitstellung einer darauf basierenden Schnittstelle, beispielsweise zur einfachen Einstellung von Kopf und Fuß einer Seite, wird dagegen spezialisierten Schnittstellenpaketen überlassen. Dokumentiert sind die Möglichkeiten von scrlayer daher im Expertenteil der KOMA-Script-Anleitung, [4].

Kombination von fancyhdr und scrlayer

Bereits während des Vortrags zu scrlayer bei der Jubiläumstagung von DANTE e.V. in Heidelberg, auf dem der Artikel [2] basiert, wurde die Frage geäußert, ob man die weitreichenden Möglichkeiten von scrlayer auch zusammen mit fancyhdr verwenden könne. Damals wurde die Frage aus dem Auditorium heraus damit beantwortet, dass dies nicht notwendig sei, da bereits scrlayer selbst die Definition von Seitenstilen erlaube und dies mit scrlayer-scrpage genauso einfach sei wie mit fancyhdr. Grundsätzlich ist dies richtig. Allerdings gibt es Anwender, die fancyhdr bevorzugen. Damit ist die Frage durchaus berechtigt.

Wie im letzten Abschnitt erwähnt, ist die Verwendung der Ebenen von scrlayer daran gebunden, dass der aktuelle Seitenstil ein mit scrlayer definierter Ebenenseitenstil ist. Dies ist bei den Seitenstilen von fancyhdr nicht der Fall. Damit ist zwar eine parallele Verwendung der beiden Pakete möglich, gleichzeitig auf derselben Seite beispielsweise den Seitenstil fancy und Ebenen von scrlayer zu verwenden, ist dagegen nicht möglich.

Erinnern wir uns jedoch kurz, dass scrlayer als Grundlagenpaket konzipiert ist, auf dem aufbauend Benutzerschnittstellen bereitgestellt werden können. Erinnern wir uns weiter, dass scrlayer-scrpage darauf aufbauend eine Schnittstelle anbietet, die weitgehend kompatibel zu scrpage2 ist, um eben dieses Paket zu ersetzen.

Es drängt sich daher die Frage auf, ob man nicht ebenso eine zu fancyhdr kompatible Schnittstelle bereitstellen kann, unter deren Oberfläche scrlayer arbeitet. Bereits beim Design von scrlayer war ich mir durchaus bewusst, dass ein solcher Wunsch entstehen könnte, und hatte von vornherein eingeplant, dass dieses Paket alles bereitstellt, um eine mit fancyhdr kompatible Schnittstelle bereitzustellen.[1]

Beim Design und der Implementierung sind zwei sehr unterschiedliche Ansätze denkbar. Zum einen könnte man, genau wie bei scrlayer-scrpage, ein komplett neues Paket schreiben, das sämtliche, dokumentierte Anwenderbefehle von fancyhdr auf Grundlage von scrlayer bereitstellt. Dieser Ansatz hätte den Vorteil, dass er sauber aufgebaut ein von der Implementierung von fancyhdr unabhängiges Paket bereitstellen würde. Die Kompatibilität mit scrlayer wäre dabei sehr hoch und man könnte optional leicht Abweichungen von fancyhdr implementieren. Änderungen an fancyhdr würden zwar möglicherweise die Kompatibilität einer solchen Implementierung gefährden, nicht jedoch deren generelle Funktion. Nachteil wäre, dass die Kompatibilität zu fancyhdr nur im Rahmen der aktuell dokumentierten Möglichkeiten gegeben wäre und etwaige Änderungen an fancyhdr gegebenenfalls nachgeführt werden müssten.

Zum anderen könnte man, wie in solchen Fällen häufig üblich, ein sogenanntes *Wrapper*-Paket schreiben. Ein solches Paket würde intern sowohl scrlayer als auch fancyhdr laden und dann nur die notwendigen Anpassungen an den (internen) Makros von fancyhdr vornehmen, um dieses auf die Beine von scrlayer zu stellen. Vorteil dieses Ansatzes wäre, dass er automatisch alle Befehle von fancyhdr bereitstellen würde. Die Kompatibilität auf Befehlsebene wäre also gegeben. Die Kompatibilität in der Funktion würde jedoch sowohl von der Implementierung des neuen Pakets als auch von fancyhdr abhängen. Als Nachteil bestünde also jederzeit die Gefahr, dass bei Änderungen an fancyhdr die Funktion des neuen Pakets nicht mehr gewährleistet wäre.

In Anbetracht der Tatsache, dass fancyhdr ein sehr stabiles Paket ist, tiefgreifende Änderungen also nicht erwartet werden, habe ich mich Mitte 2018 entschlossen, den zweiten Ansatz zu verfolgen. Nachdem ich mir die Implementierung näher angeschaut habe und dabei auf die im ersten Abschnitt dieses Artikels erwähnte Tatsache gestoßen bin, dass alle Seitenstile von fancyhdr letztlich auf demselben, internen Seitenstil @fancy basieren und alle Anweisungen von fancyhdr darauf abzielen, genau diesen Seitenstil zu konfigurieren, drängte sich die Idee auf, dass nur eben dieser durch einen auf scrlayer basierenden Ebenenseitenstil ersetzt werden

müsste, um die Funktion von fancyhdr zu gewährleisten und gleichzeitig die Ebenen von scrlayer verwenden zu können.

Da fancyhdr sehr deutlich zwischen dem Kopf und dem Fuß, jedoch nicht sehr stark zwischen den drei Elementen jeweils von Kopf und Fuß unterscheidet, habe ich beschlossen, für Kopf und Fuß eigene Ebenen zu definieren. Darüber hinaus unterscheidet fancyhdr zwischen linken und rechten Seiten, wie das auch schon der LaTeX-Kern tut. Daraus ergeben sich dann insgesamt vier Ebenen:

```
\DeclareNewLayer[%
  background,oddpage,
  head,
  contents={\hb@xt@ \layerwidth{%
      \f@nch@head\f@nch@Oolh\f@nch@olh
      \f@nch@och\f@nch@orh\f@nch@Oorh}}
]{fancy.head.odd}
\DeclareNewLayer[%
  background,evenpage,
  head,
  contents={\hb@xt@ \layerwidth{%
      \f@nch@head\f@nch@Oelh\f@nch@elh
      \f@nch@ech\f@nch@erh\f@nch@Oerh}}
]{fancy.head.even}
\DeclareNewLayer[%
  foreground,oddpage,
  foot,
  contents={\hb@xt@ \layerwidth{%
      \f@nch@foot\f@nch@Oolf\f@nch@olf
      \f@nch@ocf\f@nch@orf\f@nch@Oorf}}
]{fancy.foot.odd}
\DeclareNewLayer[%
  foreground,evenpage,
  foot,
  contents={\hb@xt@ \layerwidth{%
      \f@nch@foot\f@nch@Oelf\f@nch@elf
      \f@nch@ecf\f@nch@erf\f@nch@Oerf}}
]{fancy.foot.even}
```

Die verwendeten internen Befehle, die mit \f@nch@ beginnen, sind dabei in fancyhdr definiert. Bis auf das zusätzlich eingefügte \hb@xt@ \layerwidth, das einer \makebox mit linksbündigem Inhalt in Ebenenbreite entspricht, kopiert die Definition der Inhalte der Ebenen genau die Definition von Kopf und Fuß von ungeraden und geraden Seiten im Seitenstil @fancy.

Basierend auf den vier Ebenen wird dann der Seitenstil \@fancy mit

```
\DeclarePageStyleByLayers[
  onselect={\def\@mkboth{\protect\markboth}},
]{@fancy}{%
  fancy.head.odd,fancy.head.even,
  fancy.foot.odd,fancy.foot.even
}
```

umdefiniert. Da alle Seitenstile von fancyhdr letztlich diesen einen Seitenstil verwenden, sind bereits damit alle von fancyhdr definierten Seitenstile auf die Verwendung von scrlayer umgestellt. Für den Seitenstil plain gilt dies jedoch nur, wenn man entweder mit

```
\pagestyle{fancyplain}
```

den in der fancyhdr-Anleitung nicht dokumentierten Seitenstil verwendet, der

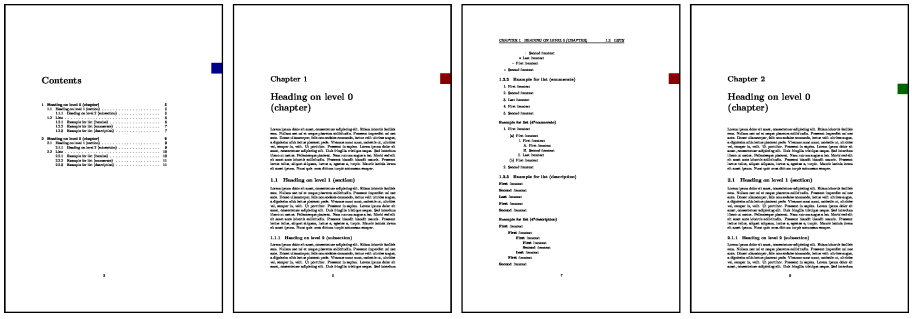

Abbildung 1. Kombination von farbigen Kapitelmarken mit `scrlayer` mit einem Seitenstil von `fancyhdr`

einerseits ebenfalls den Seitenstil `fancy` aktiviert und andererseits den Seitenstil `plain` umdefiniert, oder aber den Seitenstil `plain` selbst mit `\fancypagestyle` oder `\DeclarePageStyleByLayers` umdefiniert.

Wozu das Ganze?

Mit dem neuen Paket `scrlayer-fancyhdr` ist es nun möglich, Ebenen wie in [2] mit einem Seitenstil von `fancyhdr` zu kombinieren. Man erhält dann beispielsweise Seiten wie in Abbildung 1 gezeigt. Gegenüber dem Originalquelltext aus [2] wurde lediglich `scrlayer-fancyhdr` anstelle von `scrlayer-scrpage` geladen und ein `\pagestyle{fancyplain}` eingefügt.

Aber auch unmittelbare Erweiterungen der Fähigkeiten von `fancyhdr` sind so möglich. Beispielsweise könnte man einfach den Seitenkopf farbig hinterlegen:

```
\documentclass[a4paper]{article}
\usepackage{scrlayer-fancyhdr}
\usepackage[svgnames]{xcolor}
\usepackage{blindtext}
\pagestyle{fancy}
\DeclareNewLayer[%
  background,head,addheight=.5ex,
  contents={\color{Coral}\rule{\layerwidth}{\layerheight}}
]{Background}
\AddLayersAtBeginOfPageStyle{@fancy}{Background}
\begin{document}
\tableofcontents
\blinddocument
```

Abbildung 2 zeigt das Ergebnis.

Beschränkungen

Das aktuelle verfügbare Paket ist noch nicht für den produktiven Einsatz gedacht. Es dient lediglich ersten Experimenten, um weitere, mögliche Probleme zu ermitteln und deren Relevanz beurteilen zu können.

Wer sich die Deklaration des Ebenseitenstils `@fancy` genauer angeschaut hat, dem wird vielleicht aufgefallen sein, dass dabei `\@mkboth` umdefiniert wird. Das führt dazu, dass dieser Seitenstil automatische Kolumnentitel an den `scrlayer`-Befehlen `\automark` und `\manualmark` vorbei aktiviert. Dies ist beabsichtigt und entspricht dem Verhalten von `@fancy` bei `fancyhdr`.

Vergleicht man in Abbildung 3 das Ergebnis eines einfachen Dokuments mit `fancyhdr` mit dem Ergebnis eines einfachen Dokuments mit `scrlayer-fancyhdr`,

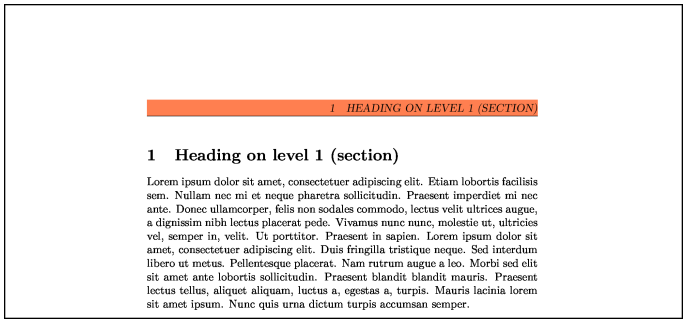

Abbildung 2. Farbig hinterlegte Kopfzeile durch Kombination von `fancyhdr` und `scrlayer`

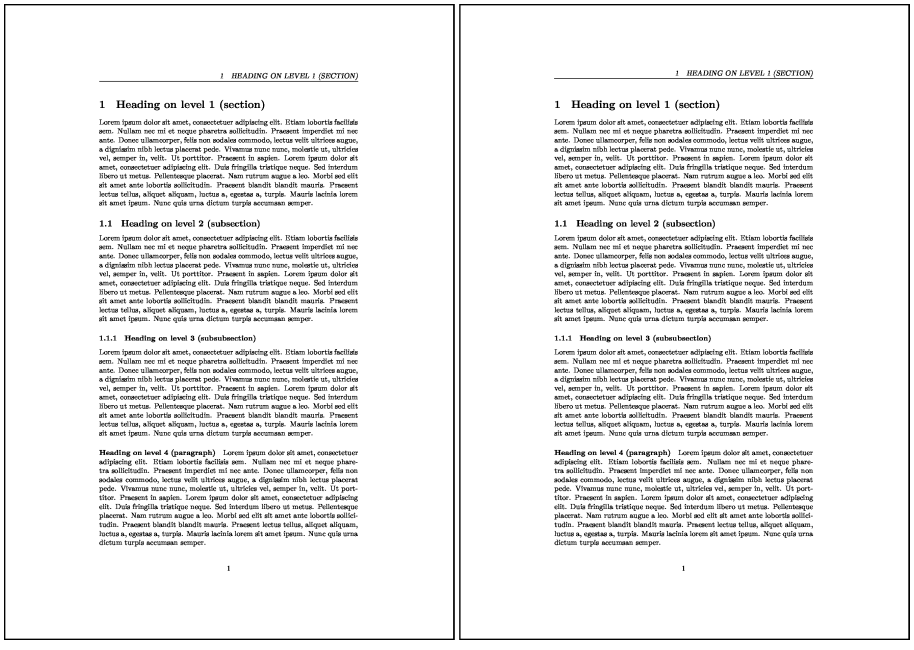

Abbildung 3. Gegenüberstellung eines einfachen Dokument mit `fancyhdr` (links) und mit `scrlayer-fancyhdr` (rechts)

so fällt eventuell auf, dass die vertikale Position des Kopfes nicht ganz identisch ist. Hier muss gegebenfalls noch nachgebessert werden. In vielen Fällen wird dieser Unterschied jedoch auch gar keine Rolle spielen.

Aktuell wirken sich Änderungen an den Ebenen von `@fancy` auf alle mit `fancyhdr` definierten Seitenstile gleichermaßen aus. Will man bestimmte Ebenen nur für einzelne Seitenstile, so muss man diese gegebenenfalls in der Definition des Seitenstils hinzufügen und auch wieder entfernen. Ob das Resultat dann immer den Erwartungen entspricht, ist noch unklar.

Ebenso wird die Kompatibilität zwischen den KOMA-Script-Klassen und `fancyhdr` durch die Verwendung von `scrlayer-fancyhdr` nicht verbessert. Dies ist auch nicht Ziel des Experiments und würde sich umgekehrt auf die Kompatibilität von `scrlayer-fancyhdr` mit `fancyhdr` auswirken.

Wie geht es weiter?

Trotz der im vorherigen Abschnitt genannten Einschränkungen kann das Experiment als Erfolg betrachtet werden. Dennoch wird von einem produktiven Einsatz derzeit noch abgeraten. Wer möchte kann aber bereits jetzt `scrlayer-fancyhdr` testen. Zum Zeitpunkt des Verfassens dieses Artikels wird dazu die aktuelle Pre-Release von [5] benötigt. Darin findet sich auch eine eigene Anleitung zu dem Paket. Es ist geplant, die aktuelle Testversion von `scrlayer-fancyhdr` auch mit der nächsten offiziellen Release von KOMA-Script zu verteilen. Von den Rückmeldungen durch die Anwender hängt es dann ab, inwiefern dieses Paket weiterentwickelt und fester Bestandteil von KOMA-Script wird.

Literatur und Software

[1] Markus Kohm: »Dokumentversion mit `scrlayer`«, DTK, 27.3 (2015), 20–24.
[2] — »Farbige, kleine Kapitelmarken am Rand mit `scrlayer`«, DTK, 27.3 (2015), 24–30.
[3] — »Firmenlogo mit scrlayer«, DTK, 27.3 (2015), 14–19.
[4] — KOMA-Script, ein wandelbares LaTeX 2$_\varepsilon$-Paket, 2018, `CTAN:/macros/latex/contrib/koma-script/doc/scrguide.pdf` (besucht am 26. 9. 2018).
[5] — KOMA-Script, The `scrlayer` interface `scrlayer-fancyhdr`, 2018, `https://komascript.de/current` (besucht am 26. 9. 2018).
[6] Piet van Oostrum: Page layout in LaTeX, 2017, `CTAN:http://mirrors.ctan.org/macros/latex/contrib/fancyhdr/fancyhdr.pdf` (besucht am 26. 9. 2018).

Notes

1. Allerdings hatte ich gehofft, dass mir die damit verbundene Arbeit jemand abnehmen würde.

Markus Kohm
komascript@gmx.info