

fluxmem 문서의 preamble

Karnes Kim

2009년 4월 2일

차 례

들어가는 말

이 문서는 `fuXmem.tex`의 코드를 설명하기 위하여 작성한다. 해당 문서는 주로 `memoir` 패키지와 `memhangul-ucs` 패키지를 사용할 때 페이지 레이아웃을 어떻게 설정할 것인지에 초점을 두고 만들어졌다. 따라서 이 글에서는 해당 문서의 `preamble`을 한 행씩 살펴보면서, 각각의 세세한 기능들을 어떻게 구현했는지 알아보려 한다.

`fuXmem` 문서는 요한 폭스(1660–1741)의 위대한 저서 『대위법 연구』(?)의 몇몇 장을 조판해본 것이다. 처음에는 `HLATEX` 문서로 작성되었던 것인데, 이번에 그것을 `memoir` 패키지를 이용한 문서로 바꾸어 본 것이다.

용지크기, 종이크기, 편집영역크기

memoir 패키지에서 페이지 레이아웃을 설정하는 것과 같은 방식으로 한다. 표준 L^AT_EX 클래스에서 이것은 geometry 패키지에 의존하던 기능이다. memoir 패키지는 아주 쉽지는 않지만 훨씬 유연한 레이아웃 설정을 가능하게 해준다.

검토할 코드는 다음과 같다.

```
1 \setstocksize{248mm}{176mm}
2 \settrimmedsize{\stockheight}{\stockwidth}{*}
3 \settypeblocksize{.8\paperheight}{.75\paperwidth}{*}
4 \setlrmargins{*}{*}{0.72}
5 \setulmargins{*}{*}{1.2}
6 \setheadfoot{\headheight}{5\onelineskip}
7 \checkandfixthelayout
8 \ifpdf
9 \fixpdflayout
10 \else
11 \AtBeginDvi{\special{pdf: pagesize height 248mm width 176mm}}
12 \fi
```

`\setstocksize`

이 명령은 용지 크기를 설정한다. 여기서는 임의의 크기인 176mm×248mm를 지정하였다. 이 크기는 보통 크라운판이라고 불리는 판형이다. 그러나 비표준 판형이기 때문에 크기를 이와 같이 별도로 지정해주어야 한다. 이 명령을 쓸 때 주의할 점은 첫번째 인자가 높이(height)이고 두번째 인자가 너비(width)라는 점이다.

`\settrimmedsize`

이 명령은 종이 크기를 설정한다. memoir 패키지에서는 용지(stock)와 종이(paper)를 구분한다. 앞의 것은 인쇄될 바탕 용지이고, 두번째 것은 편집에서 사용할 종이 크기이다. 예를 들면, A5 사이즈의 문서를 A4 용지에 인쇄할 수 있는 것이므로, 이 둘을 구분하는 것은 매우 편리하다. 그리고, 만약 이 둘이 다른 경우에는 자를 선 부분을 표시할 수 있게 해준다. 그러나 여기서는 용지와 종이를 같은 것으로 보고, trim에 대해서는 생각하지 않기로 하였다. 또, 이 명령의 인자를 주는 방법은 매우 다양한데, 여기서는 `\paperheight`와 `\paperwidth`를 직접 지정하는 방법을 사용하였다. 가로×세로의 비로 지정하는 방법도 있으므로, `memman.pdf` 문서 (?)를 참고하라.

만약 원한다면 A4 페이지를 기준으로 자르는 위치를 표시하게 할 수 있다. 이 문서가 그렇게 작성되었는데, 이 경우에는 `\setstocksize`를 설정하지 않고 `[a4paper]` 패키지 옵션을 이용하는 쪽이 낫다.¹ 그리고 이 경우 trim 영역을 표시해주려면 `[showtrims]` 패키지 옵션을 선언한다. 그러므로, 이 경우에는 다음과 같은 모습을 취하게 된다. 이것은 이 문서의 레이아웃 설정 시작부분이다.

```
\documentclass[a4paper,showtrims]{memoir}
\settrimmedsize{248mm}{176mm}{*}
\setlength{\trimtop}{\stockheight} % \trimtop = \stockheight
\addtolength{\trimtop}{-\paperheight} % \trimtop = \stockheight - \paperheight
\setlength{\trimedge}{\stockwidth}
\addtolength{\trimedge}{-\paperwidth}
\settrims{.5\trimtop}{.5\trimedge}
\trimlmarks
```

일반적으로는 이 방식으로 문서를 작성하는 것이 옳다. 그러나 여백없는 PDF를 타겟으로 하는 경우라면, 처음에 예시하였듯이 `\setstocksize`에서 용지 크기를 정한 다음 trimmed size는 stocksize와 동일하게 하는 방식도 괜찮으리라 생각한다.

`\settypeblocksize`

이 명령은 “편집영역”을 설정한다. 편집 영역은 실제로 문단이 놓이는 영역을 가리킨다. 이 영역의 크기는 `\textwidth × \textheight`로 지정되는데, 2단 문서에서는 이 영역을 다시 두 부분으로 쪼개서 쓰게 된다.

이 명령도 마찬가지로, 비례로 지정하는 방법과 값을 직접 지정하는 방법이 있으나, 여기서는 `\paperheight`와 `\paperwidth`에 대한 상대값을 각각 세로, 가로의 값으로 지정하는 방법을 사용하였다.

`\setlrmargins`

이 명령은 종이(paper)의 편집영역을 제외한 좌우의 여백(margin)을 설정하기 위해 사용하였다.

위의 `\settypeblocksize` 명령을 쓰지 않고 여백을 한꺼번에 설정할 수 있게 하는 `\setlrmarginsandblock` 명령도 있는데, `\setlrmargins`를 사용하려면 미리 `\settypeblocksize`로 `\textwidth`의 크기를 정해두어야 한다. 이 상태에서 왼쪽 여백과 오른쪽 여백의

¹trimmark를 표시해주는 데는 주로 `crop` 패키지가 쓰였다. 그러나 `memoir` 패키지에서는 별도로 `crop` 패키지를 쓰지 않고 이와 같이 설정하여 trimmark를 표시할 수 있다.

비례값을 지정하는 방법을 여기에서 적용하였다. 세번째 인자의 값이 1.0이면 좌우 여백이 같다. 반면 1보다 크다면 좌측 여백보다 우측 여백이 더 길어진다.

여기서는 0.72를 선택하였으므로, (우측 여백 길이/좌측 여백 길이)가 0.72가 되도록 좌측 여백의 길이를 더 길게 잡도록 하였다.²

`\setulmargins`

이 명령은 상하 여백을 설정하는 것이다. 앞서 `\settypeblocksize`에 의해서 `\textheight`가 이미 주어졌기 때문에, 되도록이면 여기서는 `\setulmargins`를 쓰는 것이 좋다. 상하 마진에 절대값을 적용하려면 `\setulmarginsandblock` 명령을 쓸 수는 있으나, 주의할 점은 `\textheight`가 주어진 상태에서 마진의 절대값을 지정하는 경우 `\textheight`와 이 설정에 의한 상하 마진을 합한 값이 `\paperheight`를 넘어서서는 안된다는 것이다. 그러므로 이미 `\settypeblocksize`를 썼다면 `\setulmarginsandblock`을 쓰는 것은 권장하지 않는다. 이 예제와 같이 일관성있게 설정하는 것이 좋다.

`\setheadfoot`

이 명령은 `\headheight`와 `\footskip` 값을 설정한다. 여기서는 `\headheight`는 그대로 두고 `\footskip` 값만 제법 벌렸다. 여기 사용된 `\onelineskip`은 `\baselineskip` 값과 동일하다.

`\checkandfixthelayout`

여기까지가 기본 페이지 레이아웃을 설정하는 대목이다. 제7행은 이상의 설정을 memoir 패키지가 검토하고 값을 유효하게 만들도록 하는 명령이다. 이 명령이 주어지면 .log 파일에 현재의 레이아웃 설정값이 기록된다. 이 문서의 경우에는 다음과 같다.

```
*****
Stock height and width: 845.04684pt by 597.50787pt
Top and edge trims: 69.70924pt and 48.36969pt
Page height and width: 705.62834pt by 500.7685pt
```

²좌측 여백이란, inner margin 또는 fore margin을 가리킨다. 우측 여백이란 outer margin 즉 spine margin을 가리킨다. 좌측 여백은 홀수쪽에서는 좌측이지만 짝수쪽에서는 우측, 책안쪽을 가리키는 말로 이해해야 한다.

```

Text height and width: 574.0pt by 375.0pt
Spine and edge margins: 72.78442pt and 52.4077pt
Upper and lower margins: 64.16321pt and 67.46513pt
Headheight and headsep: 12.0pt and 18.06749pt
Footskip: 60.0pt
Columnsep and columnseprule: 10.0pt and 0.0pt
Marginparsep and marginparwidth: 7.0pt and 117.0pt
*****

```

제8행에서 제12행까지는 PDF 만들기를 위한 추가 설정이다. 이 설정을 해주면 그 크기에 따라서 PDF를 잘라서 만든다. `\ifpdf..\else..\fi` 구문은 PDF_{La}T_EX이 실행되고 있는가를 체크하여 만약 PDF_{La}T_EX이 실행되고 있으면 `{true}`가 실행되고 그렇지 않으면 `\else` 이하가 실행되도록 한다. 임의의 PDF 사이즈를 정한 경우라면, 반드시 11행의 설정을 해주어야 DVIPDFMx가 PDF 자르기를 제대로 하므로 주의한다.

memhangul-ucs 관련 설정

이제 `memhangul-ucs` 패키지를 로드할 차례이다. 검토할 코드는 다음과 같다.

```
13 \ifpdf
14 \usepackage[pdftex]{graphicx}
15 \usepackage{ucshyper}
16 \hypersetup{%
17 % bookmarks=true,%
18 plainpages=false,%
19 }
20 \else
21 \usepackage[dvipdfm]{graphicx}
22 \usepackage[dvipdfm]{hyperref}
23 \hypersetup{%
24 CJKbookmarks=true,%
25 plainpages=false,%
26 colorlinks=true,%
27 }
28 \AtBeginDvi{\special{pdf: tounicode UTF8-UCS2}}
29 \fi
30 \usepackage{memhfixc}
31
32 \usepackage[interworddefault,nonfrench,gremph]{memhangul-ucs}
33 \xspaceskip=.6em plus .1em minus .05em
34 \setInterHangulSkip{0pt}
35 \noadjustquotespacing
36 \adjustfloatfnspaceing
```

제13행에서 제30행까지는 `hyperref` 패키지를 로드하는 부분이다. 물론 `graphicx` 패키지도 함께 처리하였다. `hyperref` 패키지를 언제 로드하느냐 하는 것은 아주 미묘한 문제이다. 어떤 패키지는 이것보다 먼저 와야 하는 것이 있는 반면, `hyperref` 패키지가 바꿔놓는 설정도 꽤 많기 때문에 되도록 나중에 로드해야 하는 경우도 있다. 여기서는 비교적 앞부분에서 처리했는데, 다른 패키지를 로드할 것이 많지 않다면 앞에서 해도 큰 문제를 일으키지는 않는다.

24행은 `DVIPDFMx`를 이용하여 문서를 작성할 때 PDF 북마크를 만드는 명령이다. 28행이 있어야 북마크가 제대로 만들어진다. 30행은 북마크를 만드는 경우 반드시 필요하다. 이 행을 쓰지 않으면 북마크에 불필요한 문자가 추가되는 경우가 있다.

32행에서 마침내 `memhangul-ucs` 패키지를 엮었다. 각 옵션에 대한 설명은 다음과 같다.

inderworddefault 한글 interword를 기본값으로 한다. 영문보다는 조금 느슨하고 interwordHWP보다는 조금 촘촘하다.

nonfrench 한글 문장에서도 nonfrench가 적용되도록 하는 것이다. 33행의 `\xspaceskip` 명령은 이 nonfrench의 경우 마침표 등 긴 문장부호 뒤의 공백을 설정하는 것이다.

gremph 이탤릭체 대신 그래픽체를 사용하도록 한다. 한글 기울임꼴이 그다지 미학적으로 좋은 느낌이 들지 않을 때 사용할 수 있는 옵션이다. 물론 문장 가운데에서 `\ungremph`를 선언하여 기울임꼴로 식자하는 것도 가능하다.

```
\setInterHangulSkip
```

한글의 자간을 설정한다. 일반적으로 0pt면 무난하다. 그러나 사용하는 폰트에 따라서 값을 조금 조정해야 할 경우도 있다.

```
\noadjustquotespacing
```

```
\adjustfloatfnspacing
```

이 두 명령은 `quote` 환경이나 `float` 및 `footnote` 안에서의 행간을 설정하는 데 쓰인다. `\no...`가 붙은 선언과 그렇지 않은 것이 있고 의미는 선언 자체가 의미하는 바 그대로다.

MusiX_{TEX} 관련 설정

memoir 패키지에서 musixtex 패키지를 쓰는 것은 쉬운 일이 아니었다. 우선 코드를 검토하자.

```
37  %%% oldfontcommands
38  \renewcommand\sffamily{}
39  \renewcommand\it{\itshape}
40  \renewcommand\rm{\rmfamily}
41  \renewcommand\bf{\bfseries}
42
43  %%% musixtex : CAUTION> memoir+memhangul-ucs+musixtex
44  %%% has some problems: lack of new \length.
45  %%% so, I modified musixtex.tex.
46  \usepackage{musixtex-small}
47  \input musixlyr
```

제37행에서 41행까지는 musixtex 패키지에서 사용하는 예전 스타일 폰트 명령에 호환성을 주기 위하여 설정한 것이다. memoir 패키지는 이 경우에 경고를 낼 수도 있다. 그리고 \sf 형태의 명령은 다른 경우에는 되도록 사용하지 않는 것이 좋다. 여기서는 musixtex을 사용하기 위하여 부득불 정의한 것이다.

musixtex.tex 파일에 상당히 많은 수정을 가했기 때문에 이름을 musixtex-small.tex 으로 바꾸고 로드하였다. 수정 부분은 주로 새로운 \dimen이나 \skip을 정의하는 부분 중에서 사용되지 않는 부분을 찾아서 잠시 꺼두는 방식으로 시행착오를 통하여 이루어졌다. 다른 악보에서는 다른 부분을 수정하여야 할 것이므로 일반에 공개할 만한 것은 못된다.

47행의 musixlyr는 가사의 자동처리를 위하여 로드하였다.

칼라와 폰트

색상 및 폰트 정의 부분이다.

```
48 %% color
49 \usepackage{color}
50 \definecolor{shadecolor}{gray}{.75}
51
52 %% fonts.
53 \SetHangulFonts{unbt}{ungt}{untz}
54 \SetHanjaFonts{unbt}{ungt}{ungt}
55 \SetGremphFonts{ungr}{ungt}
56 \usepackage{txfonts}
```

memoir에는 색상과 그래픽 관련 정의는 없다. 그러므로 칼라를 사용하려면 `color` 패키지를 로드하여야 한다. 원한다면 `xcolor` 패키지 등을 사용하여 다음과 같이 하는 방법도 있을 것이다.

```
\usepackage{color}
\usepackage[dvipsnames]{xcolor}
```

이것은 필자가 즐겨 쓰는 방식이다. 각 패키지의 사용법에 대해서는 패키지 문서를 참고하라.

제50행은 `shaded` 환경을 위하여 미리 `shadecolor`라는 칼라를 정의해두는 부분이다. 이 칼라는 나중에 `shaded` 환경이 불리면 배경색으로 사용된다. `framed` 환경이나 `shaded` 환경은 `framed` 패키지에 있는 것인데, `memoir` 패키지가 이 패키지를 포함하고 있기 때문에 별도로 `\usepackage`해줄 필요는 없다.

제52행에서 56행은 폰트 설정 부분이다.

```
\SetHangulFonts
```

```
\SetHanjaFonts
```

이 명령은 문서 본문의 한글 글꼴과 한자 글꼴을 설정한다. 이 두 명령은 `preamble`에서만 사용할 수 있다. 세 개의 인자를 취하는데, 각각 `\rmfamily`, `\sffamily`, `\ttfamily`에 대응할 글꼴을 지정해준다. 이 명령을 따로 사용하지 않으면 은글꼴이 사용되는데, 은글꼴로 충분하다면 그대로 사용하도록 하라. 위의 설정은 필자가 개인적으로 조성한 윤폰트의 글꼴 조합이므로, 범용으로는 쓸 수 없는 명령이다. 이런

식의 사용자 글꼴을 만드는 방법은 `ttf2hlatexfont.pl` 스크립트를 이용하는 것이 가장 간편하다.

한글 및 한자 글꼴 지정과 관련해서 주의해야 할 것은 한자 글꼴의 문제이다. 일부 한글 트루타입은 한자 영역이 없는 경우가 있으므로, 그런 글꼴을 사용했을 때 에러를 내는 수가 있다. 또, 한글과 한자가 서로 다른 글꼴이 지정되었을 때 한글 글꼴에는 한자 영역이 없을 경우, `faketfm`을 만들어주어야 하는 경우도 있다. 이런 글꼴 관련 세세한 문제는 전적으로 사용자의 책임이며, 따라서 폰트를 되도록 호환가능한 것으로 사용하여 문서를 작성하는 것이 가장 안전한 방법이 될 것이다.

55행은 `memhangul-ucs`에 `[gremph]` 옵션을 주었을 때 의미가 있다. 즉 기울인 글꼴 대신 식자할 폰트를 지정해주는 것이다. 두 개의 인자를 취하는데, 앞의 것은 한글 글꼴이고, 뒤의 것은 한자 글꼴이다.

56행은 영문 글꼴을 지정하고 있는 것이다. 여기에서는 `txfonts`를 설정하였다. 만약 영문 글꼴 설정이 없으면 EC 폰트가 사용된다.

Part, Chapter, Section 설정 및 페이지 스타일

여기서부터 이 문서의 특징 가운데 하나인 장절명령 설정을 알아본다.

```
57  %% part
58
59  \aliaspagestyle{part}{empty}
60  \renewcommand\hparttitlehead{\ \thepart.}
```

Part는 새로운 페이지를 만들면서 part 페이지 스타일을 적용한다. 처음 값은 plain과 동일한 모양이기 때문에 페이지의 바닥 중앙에 페이지 번호가 찍힌다. 이것을 없애기 위해서 part 페이지 스타일을 empty로 설정하는 것이 59행이다.

`\hparttitlehead`

이 명령은 TOC에 적어줄 Part의 headtitle을 보존하고 있다. `\renewcommand`로 변경한다. `memhangul-ucs` 패키지에서의 기본값은 “제 I 편”과 같은 형식인데, 이 문서에서는 단순히 “I.” 형식으로 하되 조금 들여쓰기를 강제하기 위하여 space를 앞에 하나 주었다. 주의할 것은, `\thepart` 자체에 ‘제’나 ‘편’과 같은 표현이 들어 있지는 않다는 점이다. 그리고 이 명령은 본문에서의 headtitle 식자와는 무관하다.

Section Style

```
61  %% sectioning
62  \setsecheadstyle{\SetAdhocFonts{ungt}{ungt}\large\bfseries}
63  \setsubsecheadstyle{\mdseries\SetAdhocFonts{ungr}{ungt}}
64  \setsecnumformat{}
65  \setsecindent{-.5\foremargin}
66  %\hangsecnum
```

섹션과 서브섹션의 스타일은 위와 같은 모양으로 설정한다. `\set<sec>indent`는 `<sec>`에 해당하는 타이틀의 들여쓰기값을 지정한다. 서브섹션의 경우라면 `\setsubsecindent`와 같이 지정하면 된다.

위의 설정으로 섹션 타이틀은 `\foremargin` 길이의 반 만큼 왼쪽으로 내어쓰기된다. 이 문서에서는 섹션 number title을 사용하지 않으므로 위와 같이 한 것이다.

\setsecnumformat은 섹션의 number title 형식을 지정한다. 여기서 이것을 비움으로써, number title이 표시되지 않도록 하였다. \section*을 쓰는 경우와 다른 점은 본문에서만 number title이 붙지 않고 TOC에 이것이 표시되도록 한다는 점이다.¹

이 문서에서는 사용하지 않았지만 \hangsecnum을 사용하면 섹션 head는 편집 영역에 왼쪽 정렬되고 섹션 number title은 왼쪽으로 내어쓰기된다. 물론 이 경우에는 별도로 \set<sec>indent를 설정하지 않아야 한다.

섹션 타이틀 내의 한글 폰트를 위하여 \SetAdhocFonts 명령이 사용되었는데, 이것은 dhucs 패키지 명령이다.

페이지 스타일

이 문서 본문에 적용될 페이지 스타일을 설계한다. 페이지 스타일을 지정할 수 있는 것은 memoir의 중요한 장점 중의 하나이고, 다양한 응용이 가능하다. 이 페이지 스타일을 MYRULED라고 부르기로 하고, Ruled 페이지 스타일을 바탕으로 설정해보기로 하자.

```

67 %% Pagestyle MYRULED
68 \coppagestyle{MYRULED}{Ruled}
69 \makepsmarks{MYRULED}{%
70   \renewcommand{\chaptermark}[1]{\markboth{##1}{}}
71   \renewcommand{\sectionmark}[1]{\markright{##1}}
72 }
73 \newlength{\MyRuledHeadwidth}
74 \setlength{\MyRuledHeadwidth}{\textwidth}
75 \addtolength{\MyRuledHeadwidth}{.5\foremargin}
76 %\addtolength{\MyRuledHeadwidth}{\spinewidth}
77 \makerunningwidth{MYRULED}{\MyRuledHeadwidth}
78 \makeheadrule{MYRULES}{\MyRuledHeadwidth}{\normalrulethickness}
79 \makeheadposition{MYRULED}{flushright}{flushright}{flushright}{flushright}
80 \makeevenhead{MYRULED}{%
81   {\sffamily\SetAdhocFonts{unjsr}{ung}\footnotesize\rightmark}%
82   }{\sffamily\footnotesize\thepage}
83 \makeoddhead{MYRULED}{%
84   {\sffamily\footnotesize\thepage}%
85   }{\footnotesize\SetAdhocFonts{unjsr}{ung}\leftmark}
86 \makeevenfoot{MYRULED}{}{\vfill FUX}
87 \makeoddfoot{MYRULED}{\vfill KARNES}{}
88
89 %% asymmetric

```

¹ 이 방법은 일반적인 문서에서는 권장할 만하다고 보기 어렵다.

```

90 \let\evensidemargin\oddsidemargin\marginparswitchfalse
91
92 %% \skip\footins
93 \setlength{\skip\footins}{1.5\onelineskip plus 5pt minus 2pt}
94
95 %% declare Main pagestyle
96 \pagestyle{MYRULED}

```

\coppagestyle

새로운 페이지 스타일을 만들 때, 이미 지정값으로 정해져 있는 스타일을 복사해 와서 수정하면 편리하다. 이 명령은 그것을 가능하게 해준다. 여기서는 새로운 스타일 이름을 MYRULED로 하고, Ruled 페이지 스타일을 복사해왔다.

수정할 생각이 없이 이름만을 바꾸어 쓰려면 \aliaspagestyle 명령을 사용할 수 있다. 이것은 앞서 59행에서 이미 설명되었다.

\makepsmarks

69–72행은 면주에 표시될 \leftmark와 \rightmark의 값을 통제하기 위하여 \chaptermark와 \sectionmark 등의 명령을 재정의하는 곳이다. 우리는 섹션의 number title이 TOC에만 나오기를 원하기 때문에 예시와 같이 정의하여 장절명령의 인자로 주어진 텍스트만으로 면주에 표시하기로 하였다.

\makerunningwidth

\makeheadrule

73–76행은 페이지 헤더의 폭을 \textwidth보다 늘리기 위하여 사용한 코드이다. 늘리고 싶은 길이를 지정한 다음 \makerunningwidth를 이용하여 헤더와 푸터의 폭을 별도로 설정한다. 실제로 이것은 표준 L^AT_EX 클래스에서는 너무나 많은 노력을 요하는 작업이었다. \makeheadrule은 헤더의 아래에 선을 긋는다.

\makeheadposition

이 명령은 만들어진 페이지 헤더를 실제로 어디에 둘 것인지를 지정하는 데 사용된다. 특히 양면조판(twoside)의 경우에는 `\textwidth`보다 긴 헤더가 편집 영역을 기준으로 `foremargin` 쪽으로 끌어낼 것인지 `spinemargin` 쪽으로 끌어낼 것인지, 아니면 양쪽 모두 반반씩 끌어낼 것인지를 지정하여야 하는데, 이 문서에서는 twoside 조판이기는 하나 홀짝수에 동일한 페이지 레이아웃을 적용하려는 것이므로 `.5\foremargin` 길이만큼 더한 다음 `flushright` 하여, 왼쪽으로 끌어내어지는 길이가 조금 더 길도록 하였다. 이 명령의 네번째와 다섯번째 인자는 footer의 위치를 설정한다. 즉, 헤더와 푸터의 위치를 이 명령 하나로 모두 조정하는 것이다.

`\makeevenhead`

`\makeevenhead`와 `\makeoddhead`, 그리고 `\makeevenfoot`, `\makeoddfoot`는 명령의 이름이 의미하는 바와 같다. 즉, 한 페이지의 면주 영역을 모두 여섯 개로 나누고, 각각 홀짝수 쪽에 대하여 설정할 수 있게 하고 있다. `fancyhdr` 패키지를 쓸 때와 기본 개념은 동일하므로 쉽게 이해할 수 있다. 장절 명령의 타이틀을 면주에 넣기 위해서 `\rightmark`와 `\leftmark`를 사용한다.

여기에서는 재미삼아서 “FUX”와 “KARNES”라는 단어를 페이지 하단의 안쪽에 찍어주었다(86-87행).

`\marginparswitchfalse`

90행은 좌우 페이지의 레이아웃이 동일하게 만드는 설정이다. 이것은 `geometry` 패키지의 `[asymmetric]` 옵션과 같은 효과가 나는데, 이렇게 하는 경우에는 `marginpar`의 위치도 조정해주어야 한다. 이를 위해서 `memhangel-ucs` 패키지에 `\marginparswitch...`

선언이 정의되어 있다. `\marginparswitchtrue`는 양면에 대하여 `marginpar`의 위치를 교체해주고, 이 기능을 끄면 홀짝수를 막론하고 항상 동일한 위치에 `marginpar`가 온다. 만약 기본 위치인 홀수쪽 `spinemargin`이 아니라 `foremargin` 쪽에다 `marginpar`를 두고 싶다면, `\reversemargintrue`를 선언한다. 이 기능들은 KTUG 게시판에서 이호재 님과의 토론 결과 얻어진 것이다.

93행에는 `\skip\footins` 길이값을 재설정하는 부분이 있다. 이것은 본문과 각주 사이를 적당히 떨어뜨려 준다. 한글 문서에서는 조금 넉넉하게 구분해주는 것이 나을 때도 있다.

* * *

그리고 나서, 기본 페이지 스타일을 우리가 방금 정의한 `MYRULED`로 선언하였다(96행).

Chapter Style

이제 chapter style을 정의할 차례이다.

```

97  %% chapter. from memman
98  \renewcommand\hchaptertitlehead{(\thechapter)}
99  \makechapterstyle{Bringhurst}{%
100   \renewcommand{\chapterheadstart}{%
101     {\let\memucsinterwordchapterskiphook\memucshangulskips}
102   \renewcommand{\printchaptername}{}
103   \renewcommand{\chapternamenum}{}
104   \renewcommand{\printchapternum}{}
105   \renewcommand{\afterchapternum}{}
106   %% next two lines added.
107   \renewcommand{\prechapternum}{}
108   \renewcommand{\postchapternum}{}
109   \renewcommand{\printchaptertitle}[1]{%
110     \raggedright\Large\scshape\bfseries
111     \SetAdhocFonts{ungr}{ungt}##1}
112   \renewcommand{\afterchaptertitle}{%
113     \vskip\onelineskip \hrule\vskip\onelineskip}
114 }
115
116 \chapterstyle{Bringhurst}

```

우리가 생각하고 있는 장 스타일은, number title이 붙지 않고, 조금 굵은 글씨로 장 제목을 표현한 다음 횡선을 하나 넣어주고 바로 본문을 시작하는 것이다. memman.pdf에서 소개되고 있는 bringhurst 예제를 거의 그대로 이용하되, 장 제목의 폰트만 조금 바꾸어주면 된다(?, 제14장). 참고로, bringhurst chapter style의 코드를 인용하면 다음과 같다.

```

%% Bringhurst chapter style
\makechapterstyle{bringhurst}{%
  \renewcommand{\chapterheadstart}{}
  \renewcommand{\printchaptername}{}
  \renewcommand{\chapternamenum}{}
  \renewcommand{\printchapternum}{}
  \renewcommand{\afterchapternum}{}
  \renewcommand{\printchaptertitle}[1]{%
    \raggedright\Large\scshape\MakeLowercase{##1}}
  \renewcommand{\afterchaptertitle}{%
    \vskip\onelineskip \hrule\vskip\onelineskip}
}

```

우선, 여기서 `memhangul-ucs` 패키지가 요구하는 바대로, `\prechapternum`과 `\postchapternum`을 추가로 재정의해준다. 여기서는 모두 비우는 것으로 한다.

`\printchaptertitle` 명령을 정의함에 있어서, `\MakeLowercase`가 쓰였는데, `\MakeUppercase`와 달리 이 명령은 한글 사용 상황에서 이따금 문제를 일으키기도 한다. 그러므로 `\MakeLowercase`는 사용하지 않기로 하고 장 타이틀의 폰트와 정렬을 109–111행과 같이 설정하였다.

제101행은 조금 미묘한 문제이다. 현재 버전의 `memhangul-ucs` 패키지는 장 타이틀에 대해서는 `\memucsinterwordchapterskip`이라는 별도의 단어 간격을 설정하고 있다. 이것은 본문에 비하여 조금 넉넉한 단어간격을 식자해주는데, 기본적으로 Huge 사이즈에 맞추어져 있기 때문에 우리와 같이 장 타이틀을 Large로 식자하려 하면 단어 간격이 너무 넓어지는 문제가 생긴다. 그래서 이것을 본문의 기본 단어간격 설정으로 되돌리기 위하여 이러한 명령을 준 것이다.

그밖의 설정

```
117 \counterwithout{table}{chapter}
118
119 \setlength{\cftpartnumwidth}{1.8em}
120 \setlength{\cftchapternumwidth}{1.8em}
```

`\counterwithout`

이 문서에서는 장과 절의 number title을 전부 제외하였기 때문에, 표나 그림의 Caption이 “그림 0.1”과 같은 형식으로 표현된다. 이것을 단순한 “그림 1”처럼 표현하기 위해서 117행의 설정을 쓴다. 이것은 `table` 카운터를 `chapter`에 동기화시키지 않고 `\thetable`을 수정해주는 역할을 한다. 카운터만을 동기화시키지 않고 `\thetable` 형식은 그대로 쓰려면 `\counterwithout*` 명령을 사용하면 된다고 한다.

119–120행은 TOC에서 Part와 Chapter의 들여쓰기 값을 조절하기 위한 것이다. 기본값에 비하여 number title이 매우 짧으므로, 예시된 것과 같이 조절한다.

이제 preamble 설정은 끝났다. 이 이후는 본문의 시작이다. 원래의 문서는 본문을 몇 개의 장으로 나누어 입력하고, 각각의 장을 include하는 방식으로 작성되었다. 마지막의 악보는 MusiX_{TeX}으로 쓰여진 코드를 그대로 가져다 썼다.

마치는 말

`fluxmem`은 그 자체로 미완성의 문서로서 예제에 불과하지만, `memoir` 패키지를 이용하여 사용자가 원하는 문서 형태를 어떻게 바꿀 수 있는지에 대한 사례가 되기를 바란다.

부 록

이 문서에서 추가된 것

이 preamble 해설 문서에서는 `fuXmem`에 다음과 같은 것이 추가되었다.

natbib 패키지

```
1 \usepackage[authoryear]{natbib}
```

참고문헌을 처리하기 위해서 `natbib` 패키지를 로드하였다. 참고문헌은 `apalike` 형식으로 하되, 다음과 같은 방법으로 입력되어 있다.

```
\bibitem[DHUCS, 2005]{DHUCS}
Dohyun Kim \& Karnes Kim.
\newblock \textit{\LaTeX\ 환경에서 Unicode 한글 문서 작성하기}.
\newblock 온택인 문서.
\newblock \texttt{dhucs} 패키지 사용자 설명서.
\newblock \url{http://faq.ktug.or.kr/mywiki/DHUCs}.
```

Trim Mark

```
2 %\setstocksize{248mm}{176mm}
3 \settrimmedsize{248mm}{176mm}{*}
4 \setlength{\trimtop}{\stockheight} % \trimtop = \stockheight
5 \addtolength{\trimtop}{-\paperheight} % \trimtop = \stockheight - \paperheight
6 \setlength{\trimedge}{\stockwidth}
7 \addtolength{\trimedge}{-\paperwidth}
8 \settrims{.5\trimtop}{.5\trimedge}
9 \trimlmarks
```

A4 용지를 기준으로 trim mark를 표시하도록 코드를 위와 같이 수정하였다. 그 결과, DVIPDFMx를 위한 페이지 사이즈 설정이 불필요해졌다.

```
10 \ifpdf
11 \fixpdflayout
12 \else
13 % \AtBeginDvi{\special{pdf: pagesize height 248mm width 176mm}}
14 \fi
```

boxedverbatim

이 문서에서는 소스를 보이기 위해 `boxedverbatim` 환경을 이용하였다. 맨 처음 나오는 `boxedverbatim`에 다음과 같이 설정되어 있다.

```
15 \bvsides\bvnumbersoutside\linenumberfrequency{1}%
16 \setverbatimfont{\normalfont\small\ttfamily}
17 \linenumberfont{\normalfont\scriptsize\sffamily\slshape}
18 \begin{boxedverbatim}
```

15행에서 17행까지는 `boxedverbatim` 환경의 모양을 설정하기 위한 것이다. 왼쪽에 라인 넘버를 붙이고, 폰트는 `\ttfamily`로 하며, 모든 행에 번호를 붙인다. 그리고 문단의 양쪽 side에 선을 그어준다.

index를 만들기 위한 local 명령들

색인을 좀더 쉽게 만들기 위해서 다음과 같은 명령을 정의하였다.

```
19 %% local commands
20 \newcommand\file[1]{%
21     \texttt{#1}\eh%
22     \index{#1}\index{파일!#1}%
23 }
24
25 \newcommand\pkg[1]{%
26     \textsf{#1} 패키지%
27     \index{#1}\index{패키지!#1}%
28 }
29
30 \newcommand\dispcomm[1]{%
31     \begin{framed}
32     \texttt{\textbackslash #1}
33     \end{framed}
34 }
35
36 \newcommand\dispcmd[1]{%
37     \texttt{\textbackslash #1}\eh%
38     \index{명령!\textbackslash #1}%
39 }
40
41 \newcommand\env[1]{%
42     \texttt{#1} 환경%
```

```

43 \index{#1}\index{환경!#1}%
44 }
45
46 \newcommand\eh{%
47     \hspace{0pt}%
48 }
49
50 \newcommand\wi[1]{%
51     \index{#1}#1%
52 }

```

Short Verbatim

\verb 형식의 명령을 쓰지 않고 간단히 verbatim을 처리하기 위해 다음과 같이 선언하였다.

```

53 \MakeShortVerb{\|}

```

참고문헌과 찾아보기

```

54 \backmatter
55 \ungremph
56
57 %% natbib, apalike style.
58 \begin{thebibliography}{}
59
60 \bibitem[DHUCS, 2005]{DHUCS}
61     Dohyun Kim \& Karnes Kim.
62     \newblock \textit{\LaTeX\ 환경에서 Unicode 한글 문서 작성하기}.
63     \newblock 온택인 문서.
64     \newblock \texttt{dhucs} 패키지 사용자 설명서.
65     \newblock \url{http://faq.ktug.or.kr/mywiki/DHUCs}.
66
67 \bibitem[Fux, 1985]{FUX}
68     Fux, Johann. (1985).
69     \newblock 나인용, 이찬매 (공역).
70     \newblock \textit{대위법 연구}.
71     \newblock 세광음악출판사.
72
73 \bibitem[Wilson, 2004]{MEMMAN}
74     Wilson, Peter. (2004).

```

```

75 \newblock \textit{The Memoir Class}.
76 \newblock 온 락인 문서.
77 \newblock (필자에 의한 일부 번역본.)
78
79 \end{thebibliography}
80
81 \renewcommand\preindexhook{%
82   웅 어 색인은 별도로 작성 하지 않았다. 여기서는 명령과 파일, 패키지에
83   대한 색인만을 실었다.
84   \vspace{\onelineskip}
85 }
86
87 \printindex

```

찾아보기를 만들기 위해서 preamble에 `\makeindex`를 선언했음은 물론이다.

51행의 `\ungremp`는 이 이후 그래픽 글꼴이 아닌 이탤릭 글꼴을 쓰겠다는 선언이다.

77행에서 81행은 찾아보기의 앞에 붙인 간단한 설명을 위한 코드이다.

fancybreak

행 나눔에 `\fancybreak`가 두어 번 쓰였다.

```

88 \fancybreak{* * *}

```

그리고 표준 L^AT_EX의 `\bigskip` 대신 다음과 같이 한 행을 띄웠다.

```

89 \vspace{\onelineskip}

```


참고 문헌

Kim, Dohyun & Karnes Kim. L^AT_EX 환경에서 Unicode 한글문서 작성하기. 온라인 문서. dhucs 패키지 사용자 설명서. <http://faq.ktug.or.kr/mywiki/DHUCs>.

Fux, Johann. (1985). 나인용, 이찬해 (공역). 대위법 연구. 세광음악출판사.

Wilson, Peter. (2004). The Memoir Class. 온라인 문서. (필자에 의한 일부 번역본.)