

KC2008 설치 후 첫 예제

KTUG Collection Team

2008년 9월 17일

KTUG Collection 2008(이하 KC2008)이 설치되었습니다.

KC2008/TeXLive는 TeXLive에서 한글을 사용하기 위한 도구를 통합한 결과물입니다.

여러분은 지금 KC2008에 포함된 Notepad++을 통해 이 문서의 소스를 보고 계십니다.

이 문서의 pdf 파일을 얻으려면 [F6]을 눌러 보십시오. 이것은 pdfLaTeX을 실행하는 것입니다. [F5]키를 누르면 한번 컴파일하여 결과를 보여줍니다. 또는 메뉴의 “플러그인-KCmenu”을 열어보시기 바랍니다. KCmenu는 plugin-KCmenu-Call KCmenu로도 열 수 있습니다.

이 문서의 dvi 파일을 얻으려면 [Alt]-[F6]로 LaTeX 컴파일 후 [Alt]-[F7]을 눌러 Dviout으로 열람하십시오. [Alt]-[F5]를 누르면 DVIPDFMx를 거쳐 pdf를 볼 수 있습니다.

다른 문의사항은 <http://www.ktug.or.kr/>의 설치운영 게시판을 이용해 주십시오.

감사합니다. 즐텍 Happy TeX'ing.

KC2008 설치 후 첫 예제

KTUG Collection Team

2008년 9월 17일

요약

KC2008이 설치되었습니다. KC2008은 \TeX LIVE가 설치되어야 잘 동작합니다. 여러분은 지금 KC2008에 포함된 `NOTEPAD++`을 통해 이 문서의 소스를 보고 계십니다.

- 이 문서의 pdf 파일을 얻으려면 `F6`을 눌러 보십시오. 이것은 $\text{\PDF}\text{\TeX}$ 을 실행하는 것입니다. `F5` 키를 누르면 한번 컴파일하여 결과를 보여줍니다. 또는 메뉴의 “플러그인-KCmenu”을 열어보시기 바랍니다. `KCMENU`는 `plugin-KCmenu-Call KCmenu`로도 열 수 있습니다.
- 이 문서의 dvi 파일을 얻으려면 `Alt-F6`으로 \TeX 컴파일 후 `Alt-F7`을 눌러 `DVIOUT`으로 열람하십시오. `Alt-F5`를 누르면 `DVIPDFMX`를 거쳐 pdf를 볼 수 있습니다.
- $\text{\Xe}\text{\TeX}$ 으로 컴파일하여¹ pdf를 볼 수도 있습니다. TrueType 폰트를 사용합니다. 이 예제 문서에서 라틴 폰트는 로만 계열로 `Palatino Linotype`, 산세리프 계열로 `Lucida Sans`, 모노스페이스 계열로 `Courier New`를 사용하도록 했습니다. 한글 폰트는 은 글꼴 TrueType을 사용하는데 명조 계열로는 바탕, 고딕 계열로는 돋움, 타자 계열로는 타자를 씁니다.

```
> xelatex first
```
- $\text{\Lua}\text{\TeX}$ 으로 컴파일하여² pdf를 볼 수도 있습니다. 이때는 라틴 폰트로 `Computer Modern` 대신 `LM OpenType` 폰트를 씁니다. 한글 폰트는 은 글꼴 TrueType을 씁니다.

```
> pdflualatex first
```
- 다른 문의사항은 <http://www.ktug.or.kr/>의 설치운영 게시판을 이용해 주십시오.

감사합니다. 즐텍 Happy \TeX 'ing.

¹ $\text{\Xe}\text{\TeX}$ 은 Jonathan Kew가 만든 오픈타입/트루타입을 이용하는 유니코드 텍 시스템입니다. $\text{\Xe}\text{\TeX}$ 은 $\text{\Xe}\text{\TeX}$ 의 \TeX 관입니다.

² $\text{\Lua}\text{\TeX}$ 은 Hans Hagen, Harmut Henkel, Taco Hoekwater가 만들고 있는 lua 스크립트 언어를 내장한 $\text{\pdf}\text{\TeX}$ 의 확장 시스템입니다. $\text{\Lua}\text{\TeX}$ 은 $\text{\Lua}\text{\TeX}$ 의 \TeX 관입니다.

1 돌다리 두드리기

최종 출력물 얻기

\LaTeX 컴파일의 최종 출력물을 무엇으로 삼을 것이냐에 따라 몇 가지 방법이 있습니다. 전에는 dvi 가 최종 출력물일 때도 있었지만 $pdf\TeX$, $CON\TeX T$ 의 등장으로 pdf 가 최종 출력물일 수도 있습니다. 대략 다음과 같이 나눌 수 있습니다. 이 글에서 우리가 컴파일하고자 하는 tex 소스의 파일 이름을 $foo.tex$ 이라 하지요.

dvi 간단하게 다음과 같이 하면 됩니다.

```
> latex foo
```

물론 옵션을 줄 수도 있습니다. 예를 들어

```
> latex -src-specials foo
```

와 같이 하면 ‘소스 스페셜’을 생성된 dvi 파일에 집어 넣습니다. 이것은 나중에 거꾸로 찾아가기(inverse search)와 같은 유용한 기능에 사용됩니다. 거꾸로 찾아가기는 dvi 의 특정 부분을 두드렸을 때 해당 부분의 tex 소스로 찾아가게 하는 기능입니다.

pdf $PDF\LaTeX$ 을 이용하여 직접 얻는 방법이 있습니다.

```
> pdflatex foo
```

$DV\IPDFMx$ 를 이용하여 dvi 를 변환하는 방법도 있습니다.

```
> dvipdfmx foo
```

그림 삽입

\LaTeX 에 그림을 넣는 방법은 다양하게 존재했고 많은 발전을 거듭했습니다. 엄밀히 말하면 \LaTeX 은 그림 포맷에 대해 모릅니다. 단지 수 많은 텍스트와 마찬가지로 그림도 가로와 세로의 길이를 지닌 상자(box)로 인식할 뿐입니다. 이렇게 상자로 인식하기 위해서 그림의 바깥 테두리의 크기(bounding box; bb)를 불러옵니다. $\backslash special$ 명령을 이용하여 넣는 방법부터 $psfig$, $epsfig$ 처럼 eps 를 겨냥한 오래된 패키지도 있었지만, 요즘은 $graphicx$ 패키지 하나만을 얻어도 일단 그림을 넣을 수 있습니다.

그러면 어떤 형식의 그림 파일을 넣을까요? 이것은 사용자가 \LaTeX 으로(혹은 $PDF\LaTeX$ 으로) 얻고자 하는 최종 출력물이 무엇이냐에 따라 몇 가지로 나눌 수 있습니다.

pdf $PDF\LaTeX$ 이나 $DV\IPDFMx$ 를 이용하여 pdf 를 얻을 때는 삽입할 수 있는 그림이 많습니다.

- PDF \TeX 을 이용할 때는 pdf와 jpg, png를 삽입할 수 있습니다. 흑백 그림인 jbig2도 이용가능합니다. eps는 직접 삽입할 수 없지만 METAPOST eps인 mps는 잘 됩니다. PDF \TeX 에서는 별도의 바운딩 박스 정보를 만들어둘 필요가 없습니다.
- DVIPDFMx를 이용할 때는 (eps를 제외한) 모든 그래픽 파일의 bb 또는 xbb를 얻어놔야 했습니다. jpg, png를 이용하려면 해당 그림의 bb를 만들어 둔 뒤에 컴파일하면 됩니다. bb를 얻는 유틸리티로는 가장 널리 알려진 EBB를 비롯, XBB 등이 있습니다.

EBB 먼저 sample.jpg 다음과 같이 EBB를 실행하면 sample.bb가 생겨납니다. (png, pdf에 대해서도 마찬가지로) 그 다음에 PDF \TeX 컴파일을 하면 됩니다.

```
> ebb sample.jpg
...
\usepackage{graphicx}
...
\includegraphics{sample}
```

XBB 이 유틸리티는 DVIPDFMx 프로젝트에 들어있는 것으로 EBB의 몇 가지 문제점을 수정하여 확장한 것입니다. sample.jpg에 대해 DVIPDFMx에 포함된 XBB를 실행하면 sample.xbb가 생깁니다. (png, pdf에 대해서도 마찬가지로) XBB를 다음과 같이 실행하면 sample.bb가 생깁니다. XBB의 실행 명령은 EXTRACTBB입니다.

```
> extractbb -x sample.jpg
...
\usepackage{graphicx}
...
\includegraphics{sample}
```

.xbb와 .bb XBB가 만들어진 목적 중의 하나가 DVIPDFM의 그림 처리 방법이 pdf \TeX 의 결과와 다른 경우가 있었다는 점을 해결하기 위한 것이었습니다. 새로 생긴 확장자 .xbb는 pdf \TeX 과 같은 결과를 얻기 위해 계산된 바운딩 박스 정보를 담습니다. 반면, 기존의 DVIPDFM과 같은 방식의 바운딩 박스를 얻으려면 EXTRACTBB를 EBB 호환되게 실행하면 되는데, 이렇게 해서 얻어지는 바운딩 박스 파일은 확장자가 .bb입니다. .bb를 얻으려면 extractbb에 -m 옵션을 주고 실행합니다. EXTRACTBB의 기본값은 .xbb입니다. 과거에 만들어진 문서의 호환성을 위해서가 아니라면 특별히 .bb를 써야 할 필요는 없을 것입니다.

dvi \TeX 으로 컴파일하면 dvi가 생깁니다. dvi에 그림 자체가 삽입될 수는 없습니다. dvi는 다만 그림의 바깥 테두리의 크기 (bounding box; bb)만 기억할 뿐입니다.

니다. 그렇지만 KC2008의 기본 dvi viewer 인 DVIOUT은 삽입된 그림을 pbm
이나 bmp로 바꾸어 흑백 또는 컬러로 보여줄 수 있습니다.

```
> latex foo
> dviout foo.dvi
```

그러면 그림을 하나 넣어볼까요? sample.pdf 입니다. (7쪽의 그림 1 참조)

```
\begin{figure}
\hangcaption
\captionnamefont{\small\sffamily}
\captiontitlefont{\small}
\centering{%
\includegraphics[width=.75\textwidth]{sample.pdf}}
\caption{ Garamond 폰트의 f-i 리거처
(출처: \protect\href{http://en.wikipedia.org/wiki/Typographical_ligature}
{\textsc{WikipediA}})}
\label{fig:ligature}
\end{figure}
```

그러나...모두 잊읍시다. 그림을 넣기 위해 앞에서 했던 bb 또는 xbb를 얻는 복잡한 과정을 모두 잊읍시다. T_EX_{LIVE} 2008을 기반으로 한 KC2008은 graphicx 패키지의 옵션으로 dvipdfmx를 지정한 후 L^AT_EX 명령 실행 시에 -shell 옵션을 추가하면 필요한 bb 또는 xbb를 직접 만들기 때문입니다.³ 또한 이렇게 하면 DVIPDFMX가 처리하는 jpg, png 등 그림의 크기가 PDF T_EX으로 처리한 것과 동일한 결과를 얻습니다. 예전에 EBB를 이용할 때는 DVIPDFMX가 PDF T_EX에 비해서 그림을 더 크게 불러오도록 처리했었다고 합니다.

```
> latex -shell foo
> dvipdfmx foo
```

³L^AT_EX을 실행할 때 extractbb라는 명령으로 처리됩니다.



그림 1: Garamond 폰트의 f-i 리거처 (출처: [WIKIPEDIA](#))

이 문단처럼 본문이 별행 수식보다 길게 끝나면

$$f(x) = \int \frac{\sin x}{x} dx \quad (1)$$

본문과 수식 사이의 수직 간격이 좀 벌어지게 설정되고, 본문이 다음 별행 수식보다 짧으면

$$f(x) = \int \frac{\sin x}{x} dx \quad (2)$$

본문과 수식 사이의 간격이 좁아집니다. 눈으로 보이나요?

수식에 관한 훌륭한 지침서가 있습니다. `mathmode.pdf` 를 읽어보세요.

```
> kctexdoc mathmode
```

표

LaTeX에서 표를 넣는 기본적인 방법은 `tabular` 환경을 쓰는 것입니다. 이때 최소한 자신이 작성하여야 할 표의 열(column)이 몇 칸인지는 알아야 합니다. 그리고 각 열에 들어갈 내용을 어떻게 정렬할 지는 알아야 합니다. 다음 간단한 예제를 보시지요.

```
\begin{tabular}{l|c|r} \hline
\textsf{左} & \textsf{中} & \textsf{右} \\ \hline
왼 & 가 & 오 \\
왼쪽 & 가운 & 오른 \\
왼쪽 정 & 가운데 & 오른쪽 \\
왼쪽 정렬 & 가운데 정렬 & 오른쪽 정렬 \\ \hline
\end{tabular}
```

左	中	右
왼	가	오
왼쪽	가운	오른
왼쪽 정	가운데	오른쪽
왼쪽 정렬	가운데 정렬	오른쪽 정렬

수평선은 `\hline`으로, 수직선은 `|`로 긋습니다. 이렇게 단순하게 작성된 표에서 열의 폭은 열에 들어간 내용 중 가장 긴 내용에 맞추어 정해집니다. 이 예제에서는 1열은 ‘왼쪽 정렬’ 2열과 3열은 ‘가운데 정렬’과 ‘오른쪽 정렬’의 내용이 가장 길군요. 매우 간단하지요? 그런데 문제는 이렇게 `c`, `l`, `r`로 열의 속성을 지정하면 해당 칸의 내용이 아무리 길더라도 자동으로 줄 바꿈이 되지 않는다는 것입니다. 다음 예제를 보시지요.


```
\begin{tabular}{l|c|r} \hline
\textsf{左} & \textsf{中} & \textsf{右} \\ \hline
왼 & 가 & 오 \\
왼쪽 & 가운 & 오른 \\
왼쪽 정 & 가운데 & 오른쪽 \\
왼쪽 정렬 & 가운데 정렬 & 오른쪽 정렬 \\ \hline
\end{tabular}
```

左	中	右
왼	가	오
왼쪽	가운	오른
왼쪽 정	가운데	오른쪽
왼쪽 정렬	가운데 정렬	오른쪽 정렬

웬디까, 이게 ? tabular 환경 나빠요. 그래서 array 패키지를 쓰면 몇 가지 인자를 써서 열의 폭을 지정할 수 있습니다.

```
\usepackage{array}
...
\begin{tabular}{b{2cm}|m{3cm}|p{4cm}} \hline
\textsf{左} & \textsf{中} & \textsf{右} \\ \hline
사나이 가슴에 불을 당겨 & 사나이 가슴에 불을 당겨 & 오른쪽 정렬 \\ \hline
\end{tabular}
```

左	中	右
사나이 가슴에 불을 당겨	사나이 가슴에 불을 당겨	오른쪽 정렬

오호라, 원하는대로 정해진 폭의 열이 생겨나고 그에 따라 해당 칸에서 내용이 줄바꿈이 되는군요. 그런데... 뭔가 이상합니다. b, m, p 인자를 주어 칸의 폭을 정하고 줄 바꿈이 되는 것은 좋은데, 내용이 수직으로 위 / 가운데 / 아래에 정렬될 뿐이군요. 아까 처음에 했던 왼쪽 / 가운데 / 오른쪽 정렬은 어떻게 된건가요? 예를 들어 내용 줄 바꿈도 하고 싶고 왼쪽 정렬을 하고 싶을 경우 말이에요.

```
\usepackage{array}
...
\begin{tabular}{>{\raggedright\arraybackslash}b{2cm}|
>{\centering}m{3cm}|>{\raggedleft\arraybackslash}p{4cm}} \hline
\textsf{左} & \textsf{中} & \textsf{右} \\ \hline
사나이 가슴에 불을 당겨 & 사나이 가슴에 불을 당겨 & 오른쪽 정렬 \\ \hline
\end{tabular}
```

左	中	右
사나이 가슴에 불을 당겨	사나이 가슴에 불을 당겨	<u>오르오르오르오르오르오르</u> 은쪽 정렬

성공입니다! 처음에 왼쪽/가운데/오른쪽 정렬에 관한 명령어를 결합하여 열에 할당하면 되는군요. `\raggedright`, `\centering`, `\raggedleft` 를 > 과 결합하여 썼더니 잘 됩니다. 기분은 좋는데 어째 모르는 명령어가 있네요. `\arraybackslash` 명령은 뭘니까? 표에 직접적인 영향을 미치는 것은 아닌 것 같은데.... 사실은 L^AT_EX의 `\raggedright`, `\raggedleft` 명령어가 표의 행 나눔 인자인 `\\` 을 재정의합니다. 그 리하여 위의 표에서 `\arraybackslash` 를 삽입하지 않으면 컴파일하는 데 애를 먹기 때문에 이를 바로잡기 위해서 삽입하였습니다. 이 명령어가 길다고요? 그럼 다음과 같이 해보세요. `\\` 대신 `\tabularnewline` 를 쓰는 것이지요.

```
\begin{tabular}{>{\raggedright}b{2cm}|>{\centering}m{3cm}|
>{\raggedleft}p{4cm}} \hline
\textsf{左} & \textsf{中} & \textsf{右} \tabularnewline \hline
사나이 가슴에 불을 당겨 & 사나이 가슴에 불을 당겨
& 오르오르오르오르오르오르 은쪽 정렬 \tabularnewline \hline
\end{tabular}
```

左	中	右
사나이 가슴에 불을 당겨	사나이 가슴에 불을 당겨	<u>오르오르오르오르오르오르</u> 은쪽 정렬

여기서는 행 나눔 명령인 `\\` 대신 `\tabularnewline` 을 썼습니다. 결과는 아까 표와 동일합니다. 이렇게 한 열 한 열의 폭을 지정하는 것은 알겠는데 전체 표의 폭을 정해줄 수는 없을까요? `tabular*` 환경을 쓰면 됩니다. 먼저 표의 전체 폭을 본문 가로 길이인 `\textwidth`로 정해볼까요? 이렇게 하면 상대적인 길이를 갖게 되어 부득이한 사정으로 문서의 크기를 바꾸어도 표의 길이에 신경 쓸 필요가 없겠네요.

```
\begin{tabular*}{\textwidth}{>{\raggedright}b{.3\textwidth}|
>{\centering}m{.3\textwidth}|>{\raggedleft}p{.4\textwidth}} \hline
\textsf{左} & \textsf{中} & \textsf{右} \tabularnewline \hline
사나이 가슴에 불을 당겨 & 사나이 가슴에 불을 당겨
& 오르오르오르오르오르오르 은쪽 정렬 \tabularnewline \hline
\end{tabular*}
```


12 KC2008 설치 후 첫 예제

```
> kctexdoc memman  
> kctexdoc memucs-manual
```