

# 한글라텍 길잡이

---

(버전 1.0.1)

殷 光 熙<sup>1</sup>

2005년 4월 22일

<sup>1</sup>koaunghi@kornet.net

## 목 차

<b>제 1 절</b>	<b>텍 – 문서 식자 시스템</b>	<b>1</b>
1.1	텍에 대해 . . . . .	1
1.2	라텍에 대해 . . . . .	1
1.3	한글 라텍에 대해 . . . . .	2
1.4	Ω에 대해 . . . . .	2
<b>제 2 절</b>	<b>한글라텍의 설치</b>	<b>3</b>
2.1	TDS — 파일의 위치 . . . . .	3
2.2	한글라텍의 모듬 명령 파일 설치 . . . . .	5
2.3	한글 글자체 설치 . . . . .	7
2.4	라텍 설치 . . . . .	9
2.5	요약 . . . . .	9
<b>제 3 절</b>	<b>한글라텍의 사용 설명</b>	<b>11</b>
3.1	한글 문서의 틀 . . . . .	11
3.2	문서의 모양새 설정 . . . . .	12
3.3	글자체 선택 . . . . .	13
3.3.1	영문 글자체 선택 방법 . . . . .	14
3.3.2	우리말 글자체 선택 방법의 개요 . . . . .	16
3.3.3	우리말 글자체의 가족 선택 . . . . .	17
3.3.4	우리말 글자체 시리즈 . . . . .	18
3.3.5	우리말 글자체의 모양과 크기 . . . . .	19
3.3.6	우리말 글자체의 애초값 . . . . .	19
3.3.7	그외 우리말 글자체의 모듬 명령 . . . . .	20
3.4	우리말 숫자 모듬 명령 . . . . .	21
3.5	문장부호 . . . . .	22
3.5.1	마침표 . . . . .	22
3.5.2	쉽표 . . . . .	23
3.5.3	따옴표 . . . . .	23
3.5.4	묶음표 . . . . .	24
3.5.5	이음표 . . . . .	24
3.5.6	드러냄표 . . . . .	24
3.5.7	안드러냄표 . . . . .	25
3.6	우리말 각주 판짜기 . . . . .	25
3.7	조사 이형태, 매개 모음, 지정사 탈락 . . . . .	27
3.8	우리말 문헌 인용 . . . . .	29
3.9	우리말 색인 처리 . . . . .	30
3.10	우리말 이름 . . . . .	31
3.11	그외의 우리말화 . . . . .	33
3.12	PDF 문서 만들기 . . . . .	33

목 차	2
<b>부록 A</b> 참고 문헌 데이터베이스	<b>35</b>
A.1 문헌 유형	35
A.2 기재 사항	36
A.3 저자 및 편집자 이름 형식	38
<b>부록 B</b> 한글화	<b>38</b>

## 요 약

라텍은 쪽 판짜기에 탁월한 기능을 가진 문서 식자 체계(document typesetting system)입니다. 한글라텍은 이런 우수한 문서 식자 체계로 한글도 쓸 수 있도록 하자는 취지에서 만들어진 라텍 꾸러미입니다. 라텍은 사용에 아무런 제한이 없이 보급되고 한글라텍 또한 한글을 쓰고자 하는 모든 이에게 별다른 조건 없이 유용한 도구로 제공됩니다.

이 안내서에서 다음과 같은 정보를 얻을 수 있습니다.

- 한글라텍 역사.
- 한글라텍 설치.
- 한글라텍 사용.
- 우리말 글꼴 선택.

이 글이 한글라텍을 사용하는 분들에게 조그마하나마 도움이 될 수 있기를 바랍니다.

한글라텍을 발표하면서 한글 글자체 작성에 도움을 주신 고려대학교 언어과학과 李基用 교수님(klee@korea.ac.kr), 문화관광부<sup>1</sup> 포스트스크립트 글자체를 제공하신 연세대학교 국문과 洪允杓 교수님, 한글을 보는 눈을 깨우쳐 주신 충남대학교의 노용균교수님<sup>2</sup>, 한글라텍의 문제점과 개선점을 지적하고 도와주신 원세연님<sup>3</sup>과 이천우님<sup>4</sup>, 우리말 글자체 선택 모듬 명령(macro)을 제안·수정·실험해 주신 이형석님<sup>5</sup>, 항상 컴퓨터에서의 한글 사용에 대해 전반적으로 도움을 주시는 신정식님<sup>6</sup>, KTUG을 운영하시는 조진환님<sup>7</sup>, 한글 글자체를 비롯하여 다방면에서 활동하시는 박원규님<sup>8</sup>, 한글라텍 윈도우 포팅에 노력을 아끼지 않으시는 김강수님<sup>9</sup> 그리고 한글라텍의 발전에 기여하신 많은 여러분께 감사의 마음을 전합니다. 특히 李基用 교수님은 한글라텍의 발전을 위하여 2GB 굳은 저장판(hard disk)을 2개 기증하셨습니다. 한글라텍은 이를 바탕으로 새로운 완성형 글자체와 UHC 글자체들을 만들 수 있었습니다. 한글라텍을 완성해 가면서 李基用 교수님께 감사합니다.

— 殷 光熙.

---

<sup>1</sup> 이전의 문화체육부

<sup>2</sup> yno@linguist.cnu.ac.kr

<sup>3</sup> sywon@sorak.kaist.ac.kr

<sup>4</sup> clee@achem2.ajou.ac.kr

<sup>5</sup> hyslee@sarang.kaist.ac.kr

<sup>6</sup> jshin@mailaps.org

<sup>7</sup> chofchof@ktug.or.kr

<sup>8</sup> wkpark@chem.skku.ac.kr

<sup>9</sup> karnes@doeun.pe.kr

## 제 1 절 텍 - 문서 식자 시스템

### 1.1 텍에 대해

텍(TeX)은<sup>10</sup> 미국 스탠포드 대학 수학과와 Donald E. Knuth 교수가 1977년 5월에 처음으로 만들었고, 그 해 여름에 Michael F. Plass 씨와 Frank M. Liang 씨가 지금의 텍의 전신을 만드는 공동 작업을 하였으며 같은 해 말에서 다음 해 초 사이에는 Knuth 교수가 다시 완성시킨 문서 식자 시스템입니다. 텍은 원래 SAIL 언어로 쓰여졌으나 1979년 초에는 Knuth 교수와 Luis Trapp Pardo에 의해 공동 개발된 Web 언어로 전환시키는 작업이 시작되었고 1979년과 1980년에 Ignacio A. Zabala 씨가 이를 완성시켰습니다. 텍은 1979년 후반기와 1980년 전반기에 걸쳐 Knuth 교수가 더욱 안정된 프로그램으로 정착시켰고 그해 9월에 TeX82 버전 0으로 발표되기 시작했습니다. 그 후 많은 사람들의 조언을 반영하여 1989년 9월에 최종의 심사를 거쳐 발표된 텍은 현재, 전문적인 자연 과학 문서를 작성하는 데 가장 좋은 문서 식자기로 인정받고 있습니다([Knu86a] - [Knu86d] 를 참고). 수학자인 Knuth 교수는 이 문서 식자 체계를 사용하여 수학 공식도 자유 자재로 쓸 수 있도록 했습니다. Knuth 교수는 텍 외에도 METAFONT를 개발하여 텍 문서를 인쇄하는데 필요한 양질의 글자체를 만들어 낼 수 있도록 하였고 이로서 텍은 출판 서적을 만드는 데 손색이 없는 문서 식자 체계로 자리를 잡게 되었습니다.

### 1.2 라텍에 대해

텍의 식자 기능은 매우 탁월하지만 일반 사용자의 측면에서는 프로그래머의 사고 능력을 요구하기 때문에 많은 노력과 경험으로 텍의 기능을 이해하여야 문서를 작성할 수 있습니다. 예를 들어 사용자는 글자체를 스스로 정의하고 문장과 문장 사이의 간격을 넓혀주는 등의 일을 일일이 손으로 처리해야 했습니다. 그래서 텍은 전문적인 프로그래머 층에서나 사용할 수 있었습니다.

그러나 미국의 전산과학자인 Leslie Lamport는 사용자와 텍의 사이를 좁히는 역할을 할 수 있도록 라텍(LaTeX)을 개발하여 쉽게 문서를 작성하게끔 기여하였습니다 ([Lam85] 를 참고). 그는 일반적인 문서 모양새를 텍의 모든 명령(macro)으로 정의함으로써, 프로그래밍의 경험이 없는 사용자도 쉽게 문서를 작성할 수 있도록 하였습니다. 사용자는 쓰고자 하는 글의 전체적인 윤곽에 맞는 양식을 지정한 후, 작성하는 문서의 내용에만 집중하면 원하는 모양새의 문서를 작성할 수 있게 되었습니다. 그리하여 사용자는 텍에 대한 전문 지식이 없어도 짧은 시간 내에 좋은 문서를 작성할 수 있게 되었습니다.

장기간이 흐르면서 라텍은 전세계적으로 많이 보급되었고 이에 따라 라텍이 해내야 하는 과제의 폭도 넓어졌습니다. 그 결과로 라텍은 근본적인 바탕을 바꾸고 국제화되어야 한다는 여론이 높아졌고 1992년부터는 라텍 버전 3 일감 팀(project team)<sup>11</sup>이 구성되어 새로운 라텍 버전 3의

<sup>10</sup>‘텍’은 그리스어  $\tau\epsilon\chi$ 에서 유래된 낱말입니다.

English words like ‘technology’ stem from a Greek root beginning with the letters  $\tau\epsilon\chi\dots$ ; and this same Greek word means *art* as well as technology. Hence the name 텍, which is an uppercase form of  $\tau\epsilon\chi$ .

Insiders pronounce the  $\chi$  of 텍 as Greek chi, not as an ‘x’, so that 텍 rhymes with the word blecchhh. It’s the ‘ch’ sound in Scottish words like *loch* or German words like *ach*; it’s a Spanish ‘j’ and a Russian ‘kh’. When you say it correctly to your computer, the terminal may become slightly moist.

*The TeXbook*에서 발췌.

<sup>11</sup>Leslie Lamport, Johannes Braams, David Carlisle, Alan Jeffrey, Frank Mittelbach, Chris Rowley, Rainer Schöpf: 이들은 [GMS94] 를 출간하였는데 이 책의 수입은 그 절반이 라텍 버전 3 개발팀에게 할당되어 이 팀을 지원

작업에 착수하였습니다. 1994년 6월 1일을 기해 라텍 버전 2 $\epsilon$ 을 공식적으로 내놓은 이 팀은 현재 (94년 11월) 김기 수준(patchlevel) 4를 발표하면서 계속 라텍 버전 3을 위해 작업하고 있습니다. Frank Mittelbach 씨와 Rainer Schöpf 씨는 [Sch94] 에서 라텍 버전 3 개발의 목표를 다음과 같이 설명하고 있습니다.

- 라텍의 사용 범위를 이전의 자연 과학 분야에서 벗어나, 각 분야의 문서 모양새에도 만족할 수 있는 모양새 파일을 제공하고 이런 여러 모양새를 간단한 방법으로 사용자가 선택할 수 있도록 한다.
- 위와 같은 모양새 파일을 자세히 설명함으로써 사용자가 이런 기본적인 문서 모양새를 바탕으로 자신의 독자적 요구를 쉽게 이룰 수 있게 한다.

### 1.3 한글 라텍에 대해

이렇게 라텍이 국제화되어 가는 동안 한국 KAIST 전산학과에서는 라텍으로 한글을 쓸 수 있도록 모든 명령을 만들고 한글 글자체를 준비하여 hlatex을 (최우형, 백운주) 보급하였습니다. ([최우형92] 를 참고) hlatex은 기존의 라텍에 한글 글자체를 추가하는 hfont.tex, 한글 문서 작성 환경을 정의하는 harticle.sty과 hreport.sty, hbook.sty 그리고 한글 KS 완성형 부호 체계를 텍의 모든 명령으로 변환해 주는 앞처리기(preprocessor) htex으로 구성되어 있습니다. 보급되는 글자체의 사용을 제한받으면서 출발한 hlatex은 그런대로 통신망을 통해 KAIST 외부의 일반에게 퍼지게 되었고<sup>12</sup>, 이를 바탕으로, 한자의 사용을 추가한 jhtex이 (殷光熙) 만들어지게 되었습니다. 여기에는 손쉽게 구할 수 있는 일본의 간지 글자체를 hlatex에 덧붙여 사용할 수 있도록 하였고 문서 모양새도 좀더 한글화 될 수 있도록 하였으며 글자체가 늘어남으로써 불가피해진 NFSS2의 도입이 이루어졌습니다.<sup>13</sup> 이 두 프로그램은 좀더 나은, 하나의 프로그램으로 일반에게 보급될 수 있고자 한글라텍으로 통합되었는데 이로써 한글라텍은 다음과 같이 변화하였습니다 (한글라텍 버전 0.92e).

- ① 라텍 버전 2 $\epsilon$ 에 의한 한글 문서 작성
- ② 한국 표준 부호 체계 및 한자 사용
- ③ 앞처리기(preprocessor)의 기능을 라텍 모든 명령으로 대체

KAIST 전산학과에서 시작된 hlatex은 다양한 글자체를 보급하여 어느 정도 변화있는 한글 문서를 작성할 수 있도록 하였지만, 글자체의 원천(source)이 공개될 수 없어서, 미리 만들어진 300dpi와 600dpi의 pk 파일만 제공되었습니다. 따라서 문서의 인쇄는 300dpi와 600dpi의 해상도를 갖는 인쇄기에서만 가능했습니다. 그래서 한글라텍 버전 0.93부터는 문화관광부에서 공개한 글자체와 포스트스크립트 type I 글자체, 그 외에도 두값본 그림(bitmap graphic)에서 추출된 외곽선 글자체가 METAFONT 원천으로 보급됨으로써 사용자는 아무런 제약 없이 한글라텍을 사용할 수 있게 되었습니다. 글자의 배치도 이전의 KAIST식인 조합 완성 혼합식을 따르지 않고 일반적인 한국 표준 완성형 방식을 지향하게 되었습니다.

### 1.4 Ω에 대해

텍의 앞날을 바라보면 우리말과 같이 평범하게 처리하기 어려운 언어권의 사용자들에게 많은 희망을 줍니다. 1997년 2월에 발표된 web2c 버전 7.0과 같은 예를 보더라도 한 문서에서 256개 하계끔 되어 있습니다.

<sup>12</sup>그 당시 hlatex의 사용은 KAIST 이내와 해외에서만 허용되었습니다.

<sup>13</sup>NFSS2는 라텍 버전 2 $\epsilon$ 의 주된 기능으로 New Font Selection Scheme의 약자입니다. 라텍의 글자체 선택 방식을 대폭 개선함으로써 그 동안 라텍의 취약점으로 남아 있던 글자체 선택 문제를 완벽히 해결한 기능입니다.

의 글자체밖에 사용할 수 없었던 장벽이 무너지고 글자체를 최대한 2000개까지 사용할 수 있게 됨으로써, 이 『한글라텍 길잡이』를 처리하는 데 힘들었던 불편이 완전히 사라질 수 있었습니다. 그러나 근본적인 문제는, 텍 자체가 7비트 부호 체계를 바탕으로 만들어져 있는 반면 한글 부호화 표준(KSC 5601-1987)은 8비트의 문자가 두 번 연속되는 이중 바이트의 체계라는 데에서 발생합니다. 텍 버전 3의 출현으로 유럽 언어와 같은 8비트 부호 체계의 처리는 해결되었지만 우리말과 같은 16비트 부호를 처리하는 데에는 아직 많은 문제점을 안고 있습니다. 이러한 문제의 근본적인 해결책은 16비트 부호 체계를 바탕으로 형성된 텍의 구현으로 이루어질 수 있습니다. 세계적인 추세도 각 나라의 모든 언어를 동시에 부호화 할 수 있는 통일된 부호화 방식을 지향하게 되었고, 국제 표준화 단체는 32비트 코드 체계인 국제 부호화 문자 세트(UCS)를 발표하였습니다. 이에 따라 텍에서 이 문자 세트를 사용할 수 있도록 하기 위해 8비트 코드 체계를 확장하여 16비트 코드 체계에 바탕을 둔  $\Omega$  텍을 대체하는 시도가 나오게 되었습니다. 그러므로  $\Omega$ 는 16비트 텍의 구현으로서, 유니코드를 처리할 수 있는 텍이라고 간단히 표현할 수 있습니다.  $\Omega$ 는 현재 Yannis Haralambous와<sup>14</sup> John Plaice가<sup>15</sup> 공동으로 개발하고 있는데 1996년 11월에 teTeX 버전 0.4를 바탕으로 시험판이 나왔었고, 1997년 11월에는 공개판으로  $\Omega$  버전 1.47이 발표되었습니다.  $\Omega$  공개판이 발표됨으로써  $\Omega$ 에서 사용될 수 있는 새로운 꾸러미(package)들이 많이 개발되어 왔고, web2c 버전 7.0부터  $\Omega$ 가 지원되기 시작하였습니다. 결국, 1998년 3월에 발표된 web2c 버전 7.2에는  $\Omega$  버전 1.5를 공식적으로 채택하여 함께 보급하게 되는 단계에까지 오게 되었습니다. 2005년 현재 web2c는 버전 7.5.4까지 나왔고 여기에는  $\Omega$  버전 1.23.2.3이 포함되어 있습니다. 이런 상황에서, 라텍 버전 3이  $\Omega$ 를 의미하게 된다면 우리 동양권에서는 더욱 바랄 나위가 없겠지만 아직 이에 대해 결정된 사실은 들어본 바 없습니다. 1996년까지도 라텍 버전 3팀은  $\Omega$ 를 별개의 텍으로 언급하였지만 국제화를 목표 중의 하나로 삼고 있는 라텍 버전 3에  $\Omega$ 의 도입은 심각히 고려되어질 것입니다. Fabrice Popineau의 마이크로소프트 윈도우용 텍 배포판이나 Thomas Esser가 만드는 teTeX 등은 모두  $\Omega$ 를 포함하고 있습니다. 이러한 상황을 감안할 때 앞으로의 텍은  $\Omega$  쪽으로 그 흐름의 향방이 결정되지 않을까 하는 희망을 갖게 합니다.

16비트 부호를 처리하는 텍은 우리말과 밀접한 관계가 있습니다. 그러므로  $\Omega$ 와 한글라텍의 관계는 밀접할 수 밖에 없습니다.  $\Omega$ 를 사용하여 우리말 부호를 처리할 때에, 지금까지 한글라텍에 커다란 문제로 남아있는, 기존의 텍을 바탕으로 한 2×8비트식 부호 처리는 간단히 해결됩니다.  $\Omega$  버전 1.5로 우리말 KS X 1001 문서를 처리할 때에 그 처리 속도는 텍으로 영문 문서를 처리하는 속도와 차이점이 거의 없습니다.  $\Omega$ 는 이제 거의 완숙한 단계에 도달하고 있고 앞으로는 텍을 완전히 대체하고 본격적으로 사용되는 방향으로 전개될 것을 희망하고 있습니다.

$\Omega$ 를 유니코드 텍이라고 한다면  $\Lambda$ 는 유니코드 라텍입니다. 한글라텍 버전 1.0부터는  $\Lambda$ 를 사용하여 UTF-8 부호화로 작성된 한글 문서를 처리할 수 있는 모든 명령 파일 및 글자체를 제공합니다.

## 제 2 절 한글라텍의 설치

### 2.1 TDS — 파일의 위치

TDS(TeX Directory Structure)는 ‘텍 파일 자료방의 구조’에 대한 규정입니다. 이 규정은 텍의 자료방 구조를 정규화함으로써 사용자 사이에서 발생할 수 있는 혼돈을 피하고자 하는 목적에서 만들어졌습니다. 이하는 2004년 6월 23일 발표된 TDS 버전 1.1을 기반으로 설명됩니다.

<sup>14</sup>Yannis.Haralambous@univ-lille1.fr

<sup>15</sup>plaice@ift.ulaval.ca

이 규정에 따르면, 텍에 필요한 파일들은 TDS의 출발점인 TEXMF 밑에 두게 됩니다. TEXMF는 텍의 원천 코드를 컴파일할 때 결정되며, 컴파일 시에 prefix를 /usr로 설정하면 운영 체제 내에서의 경로 /usr/share/texmf가 TEXMF로 결정됩니다. 이미 설치된 텍 운영 체제에서도 texmf.cnf 파일에서 TEXMF를 다시 설정할 수 있습니다. TDS를 준수할 경우, 서로 다른 텍 배포판에서도 최상위 자료방인 TEXMF의 경로만 다를 뿐이고 그 이하의 자료방 구조는 같습니다.

‘최상자료방’은 텍 운영 체제의 구성 요소로서 tex (텍 파일), fonts (폰트 관련 파일), metafont (폰트가 아닌 METAFONT 파일), metapost (METAPOST 파일), bibtex(BIBTEX 파일), scripts (운영 체제 독립적인 실행 파일), doc (사용자용 문서 파일), source (원천 파일), <구현> (emtex, vtex, web2c와 같은 텍 구현 관련 파일), <프로그 램> (mft, dvips와 같은 프로그램에 따른 입력 및 설정 파일) 등등이 해당됩니다.

텍 체계를 구성하는 모든 명령 파일의 위치는 다음과 같은 계층 구조를 갖습니다.

TEXMF/tex/틀/꾸러미

‘틀’은 텍의 용어로 \dump 되어지는 것들을 말합니다. 한글라텍은 라텍 틀을 쓰므로 자료방의 이름은 latex이 됩니다. ‘꾸러미’는 이름 그대로 텍의 꾸러미 이름을 가리킵니다. 꾸러미를 구성하는 파일이 하나밖에 없을 때에 자료방은 misc가 됩니다. 여러 파일로 구성되어 있을 때는 꾸러미 이름에 따른 적당한 이름의 자료방을 마련합니다. 자료방의 파일의 종류가 많을 경우에는 하위자료방을 따로 만듭니다. 한글라텍의 경우에 이 자료방의 이름이 hlatex입니다.

한글라텍은 파일의 수, 특히 글자체 파일의 수가 많아서 한글라텍 전체의 뿌리 (TEXMF)를 달리 하자는 의견이 있습니다. 일반적으로 지역 파일은 TEXMFLOCAL로 대변되는 자료방 (teTeX의 경우에는 /usr/share/texmf-local, MikTeX의 경우에는 c:\localtexmf)를 뿌리로 하지만 한글라텍의 관리를 용이하게 하기 위해서는 TEXMFHANGUL을 뿌리로 정할 수 있습니다. 자료방의 이름은 texmf-hlatex을 제안합니다. 텍 운영 체제에 이와 같은 설정을 추가하기 위해서는 texmf.cnf 설정 파일을 편집하여야 합니다. 텍 운영 체제에서 현재 사용하고 있는 설정 파일의 위치는 다음의 명령으로 찾을 수 있습니다.

kpsewhich texmf.cnf

운영 체제 관리자의 권한으로 이 파일을 편집기로 열어서 적당한 곳에 (예를 들어 TEXMFMAIN이 설정된 줄 다음에) 다음과 같은 줄을 끼워 넣습니다.

TEXMFHANGUL = /usr/share/texmf-hlatex

그리고 TEXMF가 정의된 줄을 찾아 다음과 같은 식으로 TEXMFHANGUL을 추가합니다.

TEXMF = { ..., !!\$TEXMFHANGUL, !!\$TEXMFMAIN, ... }

이에 따라 한글라텍의 모든 명령(macro) 파일들은

texmf-hlatex/tex/latex/hlatex

에 설치됩니다.

텍에 동반되는 프로그램이나 PDFTEX이나 Ω와 같이 기능이 확장된 프로그램들이 많습니다. 이와 같은 프로그램에 속하는 입력 파일들은 최상자료방인 tex 밑에 듭니다. 따라서 Λ 모든 명령 파일들은

texmf-hlatex/tex/lambda/hlatex



에 설치됩니다.

글자체는 다음과 같은 규정에 의해 설치할 것을 요구합니다.

```
TEXMF/fonts/영식/제공자/글자체
TEXMF/fonts/{enc,lig,map}/학위경로
```

‘영식’은 글자체의 종류를 가리킵니다. 즉, source, tfm, ofm, vf, ovf, type1, afm 등이 이에 해당합니다. 글자체의 부호화나 연자 혹은 글자체 이름 대응 파일들은 각각 enc, lig, map ‘영식’ 자료방에 평행한 자료방에 따로 둡니다. ‘제공자’의 대상은 글자체를 만드는 업체나 만든이 등입니다. adobe 회사의 type1 글자체 (adobe), bitstream 회사의 type1 글자체 (bitstream), D. Knuth 교수의 METAFONT 글자체 (public), J. Knappen 씨의 METAFONT 글자체 (jknappen) 등을 예로 들 수 있습니다. 한글라텍의 경우에 ‘제공자’는 uhc입니다. ‘글 자체’는 글자의 모양에 따른 이름입니다. adobe 회사에서 만든 글자체에는 times, helvetic 등의 이름이 사용되고 D. Knuth 교수가 텍 운영 체계를 위해서 만든 글자체의 경우는 cm, ec 등의 이름이 사용되며 한글라텍에서 제공하는 글자체에서는 myoungjo, gothic 등의 이름이 사용됩니다.

이에 따라 한글라텍의 글자체들은 다음의 자료방에 위치하게 됩니다.

```
texmf-hlatex/fonts/영식/uhc/{myoungjo,gothic,...}
texmf-hlatex/fonts/map/hlatex
```

이하의 소절에서는 위와 같은 TDS에 의거하여 한글라텍을 설치하는 방법을 안내합니다.

## 2.2 한글라텍의 모듬 명령 파일 설치

한글라텍의 각종 모듬 명령 파일은 KTUG (Korean TeX User Group)의 ftp 서버 혹은 CTAN (Comprehensive TeX Archive Network)의 다음 URL에서 가져올 수 있습니다.<sup>16</sup>

```
ftp://ftp.ktug.or.kr/pub/ktug/hlatex/HLaTeX-1.0.1.tar.gz
ftp://ftp.dante.de/tex-archive/language/korean/HLaTeX.tar.gz
```

파일의 압축과 묶음을 풀고 필요한 파일을 다음과 같이 해당하는 자료방에 복사함으로써 설치됩니다.

- 한글라텍 모듬 명령 파일:

원천 자료방: latex

파일: hangul.sty, euc-kr.tex, hfont.tex, hfont.sty, josa.tab, hsw\*.fd, hww\*.fd, hhw\*.fd

목표 자료방: texmf-hlatex/tex/latex/hlatex

- 한글 $\Lambda$  모듬 명령 파일:

원천 자료방: lambda

파일: u8hangul.tex, uhc-test.tex, uhc\*.fd

목표 자료방: texmf-hlatex/tex/lambda/hlatex

- 한글 $\Lambda$   $\Omega$ TP 파일:

원천 자료방: lambda

파일: hlatex.otp, hlatex.ocp

목표 자료방: texmf-hlatex/omega/{otp,ocp}/hlatex

<sup>16</sup>CTAN에서는 자료방 안에서 버전을 확인하세요.

- 한글라텍에서 halpha.bst를 사용할 때 정의되는 한글 이름 파일:
  - 원천 자료방: contrib
  - 파일: hbibname-k.tex
  - 목표 자료방: texmf-hlatex/tex/latex/hlatex/
- 한글 $\Lambda$ 에서 halpha.bst를 사용할 때 정의되는 한글 이름 파일:
  - 원천 자료방: contrib
  - 파일: hbibname-u.tex
  - 목표 자료방: texmf-hlatex/tex/lambda/hlatex/
- 한글 참고 문헌 양식 파일:
  - 원천 자료방: contrib
  - 파일: halpha.bst
  - 목표 자료방: texmf-hlatex/bibtex/bst/hlatex
- 참고 문헌 한글 이름 정의 파일:
  - 원천 자료방: contrib
  - 파일: hbname-k.tex, hbname-u.tex
  - 목표 자료방: texmf-hlatex/tex/{latex,lambda}/hlatex
- 한글 색인/어휘 양식 파일:
  - 원천 자료방: contrib
  - 파일: contrib/hind.ist, contrib/hglo.ist
  - 목표 자료방: texmf-hlatex/makeindex
- 한글 참고 문헌/색인 생성 파일: 유닉스 운영 체제에서는 contrib 자료방에 있는 원천 파일 hmakeindex.c와 hbibtex.c를 실행 프로그램으로 번역하여 운영 체제의 실행 프로그램 자료방에 복사합니다.
 

```
gcc hmakeindex.c -o hmakeindex
gcc hbibtex.c -o hbibtex
```

 윈도우즈 운영 체제에서는 contrib 자료방에 있는 hmakeindex.exe와 hbibtex.exe를 실행 프로그램 자료방에 복사합니다.
- 한글라텍에 적당하게 수정된 라텍 tools 패키지:
  - 원천 자료방: contrib
  - 파일: khyper.sty, showhkeys.sty, showhtags.sty
  - 목표 자료방: texmf-hlatex/tex/latex/hlatex
- 한글 문서에서 흔히 사용되는 각주 판짜기 양식 파일:
  - 원천 자료방: contrib
  - 파일: hangulfn.sty, hfn-k.tex, hfn-u.tex
  - 목표 자료방: texmf-hlatex/tex/{latex,lambda}/hlatex

hangul.sty은 우리말 문서의 모양새 파일로서 모든 한글라텍 문서에 사용되는 양식 파일입니다. hangul.sty은 라텍과  $\Lambda$ 가 사용될 때를 구분하여 각각 euc-kr.tex과 u8hangul.tex를 읽습니다. euc-kr.tex은 EUC-KR 부호화 문자를 처리하기 위해 hfont.tex을 읽어 들입니다. hfont.sty은 hangul.sty 대신에 사용될 수 있으며 우리말 문자를 처리하는 기능만 이용하는 모양새 파일이며 hangul.sty에서 제공하는 다양한 한글 문서 처리 양식이 사용되지 않고 기존의 영문 문서의 처리 양식이 사용됩니다. josa.tab은 자동 조사 처리를 위해 euc-kr.tex이

읽어들이는 파일입니다. 한글라텍에서 제공하지 않는 글자체의 애초값을 추가로 설정하거나 기존의 설정을 바꾸기 위해서는 `hfont.cfg`에 설정 사항을 기록하여 텍 운영 체제에 설치하면 `hfont.tex`이 자동으로 설정을 반영합니다.

한글라텍에 사용되는 글자체는 상징 문자와 한글 그리고 한자로 나뉘어져 있으며 상징 글자를 정의하는 파일이 `hsw*.fd`이고 한글 글자를 정의하는 파일이 `hws*.fd`이며 한자 글자를 정의하는 파일이 `hhs*.fd`입니다.

한글 $\Lambda$ 를 사용할 경우에는 `u8hangul.tex` 파일이 UTF-8 부호화 문자를 처리하기 위한  $\Omega$ TP ( $\Omega$  Translation Process) 파일인 `hlatex.ocp`를 가동합니다. `hlatex.otp`는 기계어로 번역되기 전의 텍스트 파일입니다. 한글 $\Lambda$ 에 사용되는 글자는 한글라텍에 사용되는 글자와 달리 상징, 한글, 한자로 나뉘지 않으며 글자체 정의 파일의 이름들은 `uhc*.fd`입니다.

`halpha.bst`는 한글 참고 문헌 데이터베이스를 사용할 때 필요한 BIBTEX 양식 파일입니다. 참고 문헌 데이터베이스를 처리할 때에는 이 한글 양식 파일에 정의된 한글 이름이 사용되는데, 한글라텍과 한글 $\Lambda$ 의 부호화 방식이 서로 다르므로 `hbname-k.tex`과 `hbname-u.tex`에 서로 다른 부호화 방식으로 한글 이름을 정의하였습니다. 한글 $\Lambda$ 에서 참고 문헌 데이터베이스를 이용할 때는  $\Lambda$ 가 보조 파일에 8비트 문자를 출력하지 못합니다. 따라서 미리 보조 파일의 한글 문자를 8비트로 변환하는 작업이 필요하며 이를 위해서는 contrib 자료방의 `hbibtex.c`를 실행 프로그램으로 번역하여 사용하여야 합니다. 윈도우즈 운영 체제에서는 미리 번역한 `hbibtex.exe`를 사용하면 됩니다.

`hind.ist`와 `hglo.ist`는 `makeindex`로 우리말 색인 및 어휘집을 작성할 때의 사용되는 모양새 파일입니다. 색인 작성 시에 문제가 되는 부분은 `\indexhead`로 구분되는 색인의 머릿글자의 처리인데 이를 위해서는 contrib 자료방에 있는 `hmakeindex.c`를 수행 프로그램으로 번역해서 사용해야 합니다. `hmakeindex`는 또한 한글 $\Lambda$ 에서 보조 파일에 출력하는 8비트 문자의 텍 표기 방식을 인식하여 `makeindex`의 문자 정렬이 제대로 수행되도록 하므로 한글라텍과 한글 $\Lambda$ 에서 동시에 사용할 수 있습니다. 윈도우즈 운영 체제에서는 미리 번역한 `hmakeindex.exe`를 사용하면 됩니다.

`hangulfn.sty`은 라텍에서 기본적으로 제공되는 각주 판짜기 형식을 수정하여 한글 문서에서 흔히 볼 수 있는 방식의 각주 판짜기를 쉽게 처리할 수 있도록 하기 위해 제공되는 패키지입니다. 이 패키지를 사용하면 각주 번호 양식을 애초값인 첨자(“1”) 혹은 괄호(“1”) 중에서 선택할 수 있고 각주면을 내어쓰기 및 들여쓰기에 의한 판짜기를 선택할 수 있습니다. 소절 3.6에 자세한 사용법이 소개됩니다. 선택 사항을 지정하는 한글 이름의 부호화에 따라 한글라텍에서는 `hfn-k.tex`을, 한글 $\Lambda$ 에서는 `hfn-u.tex`을 사용합니다.

### 2.3 한글 글자체 설치

한글라텍의 각종 글자체는 KTUG (Korean TeX User Group)의 ftp 서버 혹은 CTAN (Comprehensive TeX Archive Network)의 다음 URL에서 가져올 수 있습니다.

```
ftp://ftp.ktug.or.kr/pub/ktug/hlatex/fonts/
ftp://ftp.dante.de/teX-archive/fonts/korean/HLaTeX/
```

글자체들은 모두 한글라텍과 한글 $\Lambda$ 에서 사용될 수 있으며 다음과 같이 tar/gzip으로 압축되어 제공됩니다.

글자체 설정	uhc-base.map, uhc-extra.map	uhc-fonts-1.0.tar.gz
기본 글자체	명조	uhc-myungjo-1.0.tar.gz
	고딕	uhc-gothic-1.0.tar.gz
	타자	uhc-taza-1.0.tar.gz
추가 글자체	그래픽	uhc-graphic-1.0.tar.gz
	궁서	uhc-gungseo-1.0.tar.gz
	신문	uhc-shinmun-1.0.tar.gz
	필기	uhc-pilgi-1.0.tar.gz
	펜글씨/펜홀림	uhc-pen-1.0.tar.gz
	봄글씨	uhc-bom-1.0.tar.gz
	옛글	uhc-yetgul-1.0.tar.gz
	자모 (명조/고딕/노벨/소라)	uhc-jamo-1.0.tar.gz
	바다	uhc-vada-1.0.tar.gz
	필기a	uhc-pilgia-1.0.tar.gz
	디나루	uhc-dinaru-1.0.tar.gz

UHC 글자체는 크게 기본 글자체와 추가 글자체로 분류되며 모두 포스트스크립트 Type1 형식의 14가지 종류가 있습니다. Type1 글자체는 afm과 pfb 형식의 파일로 구성되는데, afm 파일이 tfm과 vf 형식의 파일로 전환되어 라텍에서 사용할 수 있게 됩니다. Λ용으로는 ofm, ovf 형식의 파일로 제공됩니다.

이 글자체들은 \fontseries 중 좁은 넓이(75%)에서 보통 넓이(100%), 확장 넓이(125%), 초확장 넓이(200%)까지의 크기를 갖고 있으며, 무게 부분은 명조체, 고딕체, 그래픽체, 필기체의 경우 보통 무게(medium)와 진한 무게(bold)가 있고, 디나루체에는 가는 무게(light)까지 있으며 그 외의 글자체에는 한 가지 종류만 있습니다. \fontshape은 모두 보통 모양과 기울인 모양으로 구성되며 기울인 모양이 이탤릭 모양을 대신합니다.

각종 글자체들은 TEXMF/fonts 자료방에서 gunzip/tar로 압축을 풀면 설치가 끝납니다.

```
gunzip -c uhc-myungjo.tar.gz | tar xvf -
...
```

포스트스크립트 UHC 글자체를 사용하여 인쇄 및 다른 후처리 작업을 하기 위해서는 포스트스크립트 글자체의 설정 작업이 필요하며 이를 위해 기본 글자체와 추가 글자체의 설정에 필요한 uhc-base.map과 uhc-extra.map 파일이 제공됩니다. 적어도 web2c 7.5.4 이상의 버전에서 이 fontmap 파일은 TEXMF/fonts/map의 하위 자료방에 설치되고 이전의 web2c 버전에서는 TEXMF/dvips의 하위 자료방에 설치됩니다. web2c 7.5.4 버전을 기준으로 할 때 이 두 fontmap 파일은 TEXMF/fonts/map/hlatex 자료방에 설치되는 것을 원칙으로 합니다.

uhc-fonts-1.0.tar.gz을 적당한 위치에서 풀고 uhc-fonts-1.0 자료방에 있는 두 fontmap 파일을 TEXMF/fonts/map/hlatex에 복사합니다. fontmap 파일이 설치된 후에는 운영 체제의 관리자로서 texhash를 실행 시켜서 ls-R 데이터베이스를 갱신하고

```
updmap-sys --enable Map=uhc-base.map
```

을 실행시키면 dvips, pdftex, xdvi, ps2pk, gsftopk, dviptfm 등과 같은 후처리 프로그램에서 한글 라텍 글자체를 사용할 수 있게 설정 됩니다. 추가 글자체도 함께 설치하였을 경우에는 uhc-extra.map에 대해서도 위와 같은 명령을 수행합니다.

## 2.4 라텍 설치

한글라텍은 라텍을 기반으로 움직입니다. 그러므로 운영 체제에 라텍이 설치되어 있어야 한글 문서를 만들 수 있습니다. 라텍은 모든 텍 운영 체제에 기본적으로 탑재되어 있습니다. 예를 들어 유닉스나 리눅스 운영 체제에는 주로  $\text{te}\text{T}\text{E}\text{X}$ 이 텍 운영 체제로 설치되고 MS 윈도우 운영 체제에는 주로  $\text{Mik}\text{T}\text{E}\text{X}$ 이 텍 운영 체제로 설치되는데 이 텍 운영 체제에는 라텍이 탑재되어 있습니다.

만일에 텍 운영 체제에 라텍이 포함되어 있지 않으면 가까운 CTAN에서는 모두가 파일 옮김 규약(ftp, File Transfer Protokol)을 통해 익명으로 /tex-archive에 있는 모든 텍 소프트웨어를 자유롭게 가져다 설치할 수 있습니다. 다음과 같이 라텍을 가져옵니다.

```
hpserv:> ftp ftp.dante.de
login: ftp
passwd: ftp
ftp> cd /tex-archive/macros/latex
ftp> bin
ftp> get base.tar.gz
ftp> get required.tar.gz
ftp> bye
```

가져온 압축 파일을 풀고 라텍을 설치합니다. 이전의 라텍 버전 209가 설치되어 있으면 먼저 이를 여벌 만들기(backup)합니다.

```
mv TEXMF/tex/latex TEXMF/tex/latex209
mkdir TEXMF/tex/latex
mkdir TEXMF/tex/latex/base
tar xzvf base.tar.gz
cd latex/base
initex unpack.ins
initex latex.ltx
cp latex.fmt texmf/web2c
cp *.sty *.tex *.fd *.clo *.cls *.def texmf/tex/latex/base
```

이 외에도 required.tar.gz을 풀면 amslatex, babel, cyrillic, graphics, psnfss, tools 하위 자료방이 생기며 각 하위 자료방의 꾸러미를 설치하는 것이 좋습니다. 설치 방법은 대부분의 경우 위에서 설치한 latex으로 .ins 파일을 돌리면 됩니다. 예를 들어

```
cd ../required/tools
latex tools.ins
```

CTAN의 자료방 /tex-archive/macros/latex 밑에는 다른 꾸러미들이 있으며 packages 및 contrib 자료방에서 필요한 꾸러미를 찾아 추가적으로 설치할 수 있습니다.

## 2.5 요약

운영 체제의 관리자로서 “파일”을 “경로”에 복사한 후 경우에 따라서 “설정”을 합니다.

- - 파일: `hanguk.sty`, `hfont.tex`, `eu-kr.tex` `hfont.sty`, `josa.tab`, `hfont.cfg`, `hsw*.fd`, `hww*.fd`, `hhw*.fd` (latex 자료방)
  - 경로: `texmf-hlatex/tex/latex/hlatex`
  - 설정: `texhash`
- - 파일: `u8hanguk.tex`, `uhc-test.tex`, `uhc*.fd` (lambda 자료방)
  - 경로: `texmf-hlatex/tex/lambda/hlatex`
  - 설정: `texhash`
- - 파일: `hlatex.otp`, `hlatex.ocp` (lambda 자료방)
  - 경로: `texmf-hlatex/omega/{otp,ocp}/hlatex`
  - 설정: `texhash`
- - 파일: `hangukfn.sty`, `hfn-k.tex`, `hbibname-k.tex`, `khypher.sty`, `showkeys.sty`, `showtags.sty` (contrib 자료방)
  - 경로: `texmf-hlatex/tex/latex/hlatex`
  - 설정: `texhash`
- - 파일: `hfn-k.tex`, `hbibname-u.tex` (contrib 자료방)
  - 경로: `texmf-hlatex/tex/lambda/hlatex`
  - 설정: `texhash`
- - 파일: `halpha.bst` (contrib 자료방)
  - 경로: `texmf-hlatex/bibtex/bst/hlatex`
  - 설정: `texhash`
- - 파일: `hind.ist`, `hglo.ist` (contrib 자료방)
  - 경로: `texmf-hlatex/makeindex`
  - 설정: `texhash`
- - 파일: 유닉스 운영 체제에서는 `hmakeindex.c`, `hbibtex.c`. (윈도우즈 운영 체제에서는 `hmakeindex.exe`, `hbibtex.exe`. (contrib 자료방)
  - 경로: 운영 체제의 실행 프로그램 자료방
  - 설정: 유닉스 운영 체제의 경우 실행 프로그램으로 번역, 윈도우즈 운영 체제의 경우 복사
- - 파일: `uhc-base.map`, `uhc-extra.map` (`uhc-fonts-1.0.tar.gz`)
  - 경로: `texmf-hlatex/fonts/map/hlatex`
  - 설정: `texhash`
- - 파일: `uhc-*.tar.gz`
  - 경로: `texmf-hlatex/fonts`에서 `tar/gzip`으로 압축과 묶음을 퉁다.
  - 설정: `texhash`
    - `updmap-sys --enable Map=uhc-base.map`
    - `updmap-sys --enable Map=uhc-extra.map`

## 제 3 절 한글라텍의 사용 설명

이 단원에서는 라텍 자체의 기본적인 사용법에 대한 설명을 피하고, 한글라텍에 관계되는 부분만을 간단히 다루려고 합니다. 라텍의 사용법에 관해서는 이에 해당하는 참고 서적을 보시기 바랍니다. 라텍의 원저자인 Lesley Lamport의 『 $\text{\LaTeX}$  — A Document Preparation System』 ([Lam85])를 참고로 하면 라텍에는 수 많은 명령들이 있음을 알 수 있습니다. 일반 사용자로서 이런 모든 명령을 전부 알아야 할 필요는 없습니다. 개인의 필요에 따라 약간의 기초 지식에, 필요할 때마다 [Lam85]를 참고하여 사용하면 어렵지 않게 라텍 문서 작성을 시작할 수 있습니다. 독일에서는 『 $\text{\LaTeX}$  Eine Einführung』이 ([Kop91]) 잘 알려져 있습니다. 이 책은 미국에서 영어로 번역되어 『A Guide to  $\text{\LaTeX}$ 』으로 출판되고 있고 ([KD99]) 한국에는 한국어 판으로도 번역되어 나와 있는 것으로 알고 있습니다. 이 책에서는 모든 명령들을 간단히 설명한 목록이 있어서 사용자가 빠른 시간 안에 그 명령의 개요를 파악할 수 있고, 지정된 페이지로 가면 더욱 상세한 내용을 완벽하게 이해할 수 있도록 짜여져 있습니다. 참고서로 가장 많이 추천되는 책입니다.

### 3.1 한글 문서의 틀

문서는

```
\documentclass[추 가 선택]{문 서 종 류}
```

로 시작합니다. 추 가 선택에는 라텍의 문 서 종 류(class)에 종속적인 추가 선택을 지정합니다.

문서 종류 지정 후에 한글라텍 꾸러미 사용을 지정합니다.

```
\usepackage[추 가 선택]{hangu1}
```

한글라텍의 추 가 선택의 종류는 다음과 같습니다.

- `hanja`: 문서의 단원명이나 날짜가 한자로 표기된다.
- `hardbold`: METAFONT에 의해 만들어지는 두꺼운 글씨체를 쓰도록 한다.
- `nojosa`: 자동 조사 처리 기능을 억제하여 텍의 기억 용량 사용을 감소시킨다.
- `판형`: 책의 크기를 지정하는 선택 사항입니다. 책의 크기는 라텍 자체의 추가 선택으로 지정되는데, 이 추가 선택에는 한국의 출판 업계에서 사용되는 판형에 해당하는 크기가 없으므로 한글라텍의 추가 선택으로 지정해야 합니다. 한국에서 사용되는 책의 판형에는 표 1과 같은 종류가 있습니다. 국전지의 크기는  $939 \times 636\text{mm}$ 이고 46전지의 크기는  $788 \times 1091\text{mm}$ 입니다.

`hangu1` 이외의 다른 꾸러미를 함께 쓰고자 할 경우에는 반점(,)으로 구분하여 이름을 쓰거나 따로 `\usepackage` 명령을 씁니다. `hangu1`의 추가 선택은 문서 종류의 추가 선택에서 지정할 수도 있고 여기서 지정할 수도 있습니다. 추가 선택의 처리 방식은 다음과 같습니다.

- `\documentclass`의 추가 선택은 전역적(global)이다. 어느 문서 종류나 꾸러미에서든지 이 추가 선택을 처리할 수 있다.
- `\usepackage`의 추가 선택은 올안적(local)이다. 그 꾸러미에서 이 추가 선택을 처리할 수 있어야 한다.

다음의 예에서 보는 것처럼

```
\usepackage[신 국 판]{hangu1,a4}
```

에서는 지정된 추가 선택 신 국 판을 두 꾸러미 `hangu1`과 `a4`가 모두 처리할 수 있어야 합니다. `a4` 꾸러미는 신 국 판이라는 추가 선택을 처리할 수 없으므로 오류가 발생하게 됩니다. 다음과 같

판형명칭	크기[mm]	사용종이	쪽수/전지	
국 판	148 × 210	국전지	32	교과서, 단행본
국 배판	210 × 297	국전지	16	잡지
국 반판	105 × 148	국전지	64	문고
탁블로이드 판	257 × 364	4x6전지	16	정보신문
사륙 판	128 × 182	4x6전지	64	문고
사륙 배판	182 × 257	4x6전지	32	참고서
사륙 반판	94 × 128	4x6전지	128	
사륙 배배판	254 × 374	4x6전지		
신국 판	152 × 225	국전지	32	단행본
대국 전 판	172 × 244	국전지		
크라운 판	176 × 248	4x6전지	36	사진집
삼십절 판	125 × 205	4x6전지	60	단행본
신서 판, 삼륙 판	103 × 182	4x6전지	80	문고
삼오 판	84 × 148	4x6전지		

표 1: 판형의 종류

이 각기 따로 지정하거나

```
\usepackage[신국판]{hangu1}
```

```
\usepackage{a4}
```

아니면 문제가 되는 추가 선택 신국판을 `\documentclass`에서 전역적 추가 선택으로 지정하면 오류가 발생하지 않습니다.

```
\begin{document}
```

로 문서를 시작하여 쓰고자 하는 글을 작성한 뒤

```
\end{document}
```

로 문서 작성을 마칩니다. (그림 1을 참조)

이제 latex으로 파일을 돌리면 됩니다.

### 3.2 문서의 모양새 설정

`\documentclass`는 전반적인 문서의 종류를 결정합니다. 문서종류에는 `article`, `report`, `book`, `letter`, `slides` 등이 있습니다. 각 문서종류는 추가선택에 의해 모양새가 달라질 수 있습니다. 즉, 종이의 규격을 설정하는 `a4paper`, `letterpaper`, ... 및 글자의 크기를 정해주는 `10pt`, `11pt`, `12pt` 등등이 있습니다. 일반 영문 라텍의 문서 종류 설정은 [Kop91, Lam85, GMS94] 에 잘 설명되어 있습니다.

위와 같이 라텍에서 제공하는 기본적인 문서 종류 이외에 추가로 꾸러미를 사용하고자 할 경우에는 `\usepackage` 명령을 씁니다. 우리말의 처리와 같이, 라텍에서 제공되지 않는 기능을 쓸 경우에 `hangu1.sty`을 쓰는 것과 같습니다. 각 꾸러미는 자체의 추가선택이 있습니다. 각종 꾸러미의 사용 방법 및 추가선택의 종류는 [GMS94] 에 자세히 설명되어 있습니다.

`hangu1.sty` 대신에 `hfont.sty`을 쓰면 문서 틀잡기가 우리말에 어울리는 한글화를 전혀 반영하지 않고 단순히 한글 및 한자, 상징 기호를 포함한 문서에서 우리말 처리가 제대로 될 수 있게 만 합니다. 추가 선택도 `hangu1.sty`과 같으나 의미상 `hanja`는 무시됩니다.



<code>\documentclass[추 가 선택]{문 서 종 류}</code>	
<code>\usepackage[추 가 선택]{hangul}</code>	← hangul 꾸러미 대신 hfont 꾸러미를 쓸 수 있다.
<code>\usepackage[추 가 선택]{꾸 러 미}</code>	← 필요에 따라 사용될 각종 꾸러미를 지정한다.
...	← 꾸러미뿐만 아니라 문자를 출력하지 않는 다른 명령들을 여기에서 선언할 수 있다.
<code>\begin{document}</code>	← 이전까지를 전문 영역(preamble)이라고 한다.
나락의 말이 중국과 달락서 한자로는 서로 통하지 아니하므로 이런 까닭으로 어리석은 백성들이 말하고 자 하는 바가 있어도 마침내 제 뜻을 능히 펴지 못하는 사람이 많으니라. 내가 이를 불쌍히 여겨 새로 스물 여덟자를 만드나니	← 문서의 작성은 TeX의 관습에 따른다. 즉, 중첩된 공간 문자는 하나로 처리된다. 줄바꿈 문자가 한번 나오면 공간 문자로 처리되나, 두번 이상 중첩되면 하나의 줄바꿈 문자 역할을 한다. 등등.
<code>\end{document}</code>	← 이 명령이 나온 후의 모든 문장은 무시된다.

그림 1: 한글라텍 문서의 기본 구조

영문 문서와 달리 우리말 문서에서는 줄간격을 특별히 취급해야 합니다. 이는 우리말 글자들이 영문자와 비교해서 모양이 복잡함으로 인해 발생하는 문제로서, 라텍의 애초값으로 우리말 문서를 만들 경우 각각의 쪽이 글자로 밀집된 인상을 받습니다. 이는 글을 읽는 독자의 시각을 분산시키는 악영향을 미치므로 다음과 같은 명령으로 줄간격을 조절하는 것이 좋습니다.

```
\renewcommand{\baselinestretch}{1.3}
\normalsize
```

이 명령은 줄간격을 지정하는 레지스터인 `\baselineskip`을 1.3배로 증가시키는 작용을 하며 `\normalsize`와 같이 글자체의 크기 바꿈 명령이 나타날 때 이러한 변경이 효력을 발휘합니다. 한글라텍에는 `\baselinestretch`의 기본값이 1.3으로 설정되어 있습니다. 이 설정은 문서의 어느 곳에서든지 사용자의 취향에 따라 변경될 수 있습니다.

### 3.3 글자체 선택

문서를 작성할 때, 어떤 특정한 부분의 글자체를 현재 쓰고 있는 글자체와 달리 하고 싶을 때가 있습니다. 글자의 두께를 진하게 한다든지, 크기를 크게 한다든지, 다른 모양을 쓴다든지, 또는 설계(design)가 전혀 다른 글자체를 쓴다든지 등등.

이러한 글자체 바꿈을 위해 라텍은 여러가지 글자체 바꿈 명령을 제공합니다. 라텍 209에서 제공됐던 `\rm`, `\sf`, `\sl`, `\it` 등등의 글자체 바꿈 명령들은 모두가 같은 차원에서 글자체를 바꾸기 때문에 여러가지 단점을 가지고 있습니다. 즉, 글자체의 설계가 `\rm`이 아닌 `\sf`를 계속 써 오다가 그 글자체의 기울인 모양인 `\sl`로 글자체를 바꾸면 바뀐 글자체가 `\sf`의 `\sl`이 아

나라 `\rm`의 `\s1`이 됩니다. 이는 이런 글자체 바꿈 명령이 일단 글자체를 애초값으로 복귀시킨 후 글자체를 바꾸기 때문입니다. 이러한 단점을 보완하기 위해 1989년부터 Frank Mittelbach와 Rainer Schöpf는 “새로운 글자체 선택 방식”(NFSS: New Font Selection Scheme)을 고안하기 시작하여 현재 라텍 버전 2 $\epsilon$ 에 채택되어 있는 NFSS2를 만들었습니다.

$\LaTeX$  한글라텍 버전 0.97부터는 HFSS(Hangul Font Selection Scheme)에 의해 우리말 글자체가 선택됩니다. HFSS는 NFSS2를 한글화한 것입니다.  
 $\Lambda$  한글 $\Lambda$ 에서는 우리말의 글자체 선택이 영문과 같이 NFSS2에 의해 처리됩니다.

### 3.3.1 영문 글자체 선택 방법

NFSS2는 글자체를 다섯가지 요소로 구분하여 이 요소들을 각각 독립적으로 선택할 수 있게 합니다. 그 다섯가지 요소는 다음과 같습니다.

**글자체 부호화(encoding)** 다른 알파벳을 쓸 때 각 나라별로 codepage가 정해져 있습니다. 라텍에서는 다른 프로그램과 마찬가지로 각 글자와 0-255 사이의 글자체 위치를 연관시킴으로써 이를 해결합니다. 라텍에서는 이를 “부호화 방법”이라 합니다.

**글자체 가족(family)** 글자체 가족은 같은 설계 원칙을 갖습니다. 가족 내에서는 크기(size)와 무게(weight), 폭(width), 모양(shape)으로 구성원이 구별됩니다.

**글자체 모양(shape)** 글자체 가족의 구성원을 구별하는 중요한 속성으로서 “곧은(upright) 모양”과 “굵은(*italic*) 모양” “기운(*slanted*) 모양” 혹은 대문자의 크기를 70%정도로 축소하여 소문자로 쓰는 “작은키(SMALL CAPS) 모양”등으로 구별됩니다. 글자체 모양에 있어서 그 중요도는 떨어지지만 “외곽형(outlined) 모양”과 “그림자(shaded) 모양” 및 “곧은 이탤릭(upright italic) 모양” 등도 글자체 모양에 속합니다.

**글자체 무게(weight) 및 폭(width)** 무게와 폭은 서로의 조합으로 하나의 글자체 속성을 형성합니다. 무게는 글자의 두께로서 결정되며 “보통 두께(medium)”, “가벼운 두께(light)”, “무거운 두께(bold)” ... 등으로 나뉩니다. “가벼운 두께”에도 “아주 가벼운 두께(extra light)”, “매우 가벼운 두께(ultra light)”, “보통 가벼운 두께(light)”, “약간 가벼운 두께(semi light)” 등등으로 분리되며 “두꺼운 두께”도 이런 식으로 세분됩니다. 폭은 “보통의 폭(medium width)”, “확장된 폭(extended width)” 등이 있습니다.

**글자체 크기(size)** 글자체 크기는 인쇄기의 점을 단위로 표현됩니다. 인쇄기의 점을 텍에서는 pt라는 단위로 나타내며 1인치는 72.27pt입니다. 라텍에서는 글자체 크기가 1.2의 승으로 구별됩니다. 이렇게 정한 배경에는 후에 축소 복사를 할 때의 편이함이 있습니다. 즉 A5 크기의 책자를 만들 경우, A4 종이에  $1.44 = \sqrt{2}$ 의 글자체 크기를 선택하여 인쇄한 후 A5 크기로 축소합니다.

글자체 선택에는 명령형과 선언형이 사용될 수 있습니다. 명령형은 `\textbf{...}`와 같이 한가지 변수를 요구하며, 선언형은 `\bfseries`와 같이 변수를 요구하지 않습니다. 선언형은 다음과 같이 환경 형태로도 쓰일 수 있습니다.

<pre> 몇몇 단어들은 \begin{bfseries} 두 겹 게 \end{bfseries} 인쇄됩니다. </pre>	⇒	<pre> 몇몇 단어들은 두겹게 인쇄됩니다. </pre>
---	---	---------------------------------

문서의 애초 글자체는 명령형 `\textnormal`이나 선언형 `\normalfont`로 선택되며, 이 명령/선언은 문서의 글자체를 초기화시킴으로써 항상 같은 모양을 선택할 수 있게 합니다.

문서의 글자체 가족에는 로만(roman)체, 산세리프(sans serif)체, 타자(typewriter)체가 있습니다. 이들은 각각 다음의 명령형과

```
\textrm \textsf \texttt
```

선언형으로

```
\rmfamily \sffamily \ttfamily
```

선택됩니다. 로만 글자체는 일반적으로 문서의 기본 글자체로 선언되어 있습니다.

문서의 글자체 시리즈(series)는 폭과 두께의 두 요소로 구성되며, 다음과 같은 명령 및 선언형으로 선택됩니다.

```
\textmd \mdseries
```

```
\textbf \bfseries
```

애초값은 `\textbf`가 확장된 두꺼운 두께(bold extended)이고 `\textmd`는 보통의 폭과 보통의 두께입니다.

글자체 모양은 애초값인 곧은 모양을 비롯하여 굽은 모양, 작은키 모양, 기울인 모양이 있는데 이들은 각각 다음의 명령/선언형으로 선택됩니다.

```
\textup \upshape
```

```
\textit \itshape
```

```
\textsc \scshape
```

```
\textsl \slshape
```

이 중에 굽은 모양과 곧은 모양 사이의 전환은 `\emph` 명령 및 `\em` 선언으로 이루어집니다.

글자체 크기 선택을 위해서는 10가지의 선언형이 사용됩니다. (명령형은 없습니다.)

<code>\tiny</code>	미소한 크기
<code>\scriptsize</code>	각본철 크기
<code>\footnotesize</code>	주석 크기
<code>\small</code>	작은 크기
<code>\normalsize</code>	보통 크기
<code>\large</code>	큰 크기
<code>\Large</code>	좀더 큰 크기
<code>\LARGE</code>	아주 큰 크기
<code>\huge</code>	거대한 크기
<code>\Huge</code>	아주 거대한 크기

위와 같은 외형 글자체 바꿈 명령들의 행동 방식은 내형 글자체 바꿈 명령에 의해 달라집니다. 예를들어

```
\renewcommand{\familydefault}{cmss}
```

와 같은 내형 글자체 바꿈 명령을 쓰면 글자체 가족의 애초값이 `cmss`(Computer Modern Sans Serif)로 바뀔으로써 주석 및 머릿말 등, 외형 글자체 바꿈 명령만으로는 바뀌지 않는 부분까지의 글자체 가족이 영향을 받습니다.

라텍에서 애초값은 다음 표와 같이 설정되어 있습니다.

내형 글자체	애초값
<code>\encodingdefault</code>	<code>OT1</code>
<code>\familydefault</code>	<code>\rmdefault</code>
<code>\seriesdefault</code>	<code>m</code>
<code>\shapedefault</code>	<code>n</code>
<code>\rmdefault</code>	<code>cmr</code>
<code>\sfdefault</code>	<code>cmss</code>
<code>\ttdefault</code>	<code>cmtt</code>
<code>\bfdefault</code>	<code>bx</code>
<code>\mddefault</code>	<code>m</code>
<code>\itdefault</code>	<code>it</code>
<code>\sldefault</code>	<code>sl</code>
<code>\scdefault</code>	<code>sc</code>
<code>\updefault</code>	<code>n</code>

### 3.3.2 우리말 글자체 선택 방법의 개요

#### 라텍을 쓸 때

HFSS는 NFSS2를 바탕으로 두고 있고 우리말 부호화의 특성에 따라 한국어 부호화 방식을 지정하는 `\hfontencoding`과 한국어 글자체 가족을 지정하는 `\hfontfamily`가 추가 되었으며 글자체 선택 방식도 우리말 글자체에 맞도록 변경되었습니다.

**글자체 부호화** 우리말의 부호화는 H로 설정되어 있고 NFSS2에서 사용되는 글자체 부호화 설정 모듬 명령 `\fontencoding`을 우리말화한 `\hfontencoding`을 씁니다. 우리말 글자체의 부호화는 사용자의 입장에서 볼 때 달리 설정할 이유가 없고, 영어 부호화 방식과의 구별에 그 주 목적이 있으며 HFSS의 작동시 우리말 글자체의 선택을 우리말에 맞게 하도록 하는 기능을 합니다.

**글자체 가족** NFSS2에서와 같이 우리말의 글자체 가족은 글자체의 설계 원칙에 따라 구분이 되고, 우리말의 특성에 의해 두 요소로 표현됩니다.

첫번째 요소는, `\ksfamily`로서 우리말의 상징 문자와 한글 및 한자를 구별합니다. 이 모듬 명령도 사용자의 입장에서는 쓸 일이 거의 없고, 주로 한글라텍의 내부 처리 과정에서 사용됩니다.

두번째 요소는 우리말의 글자체 가족을 나타내는 `\hfontfamily`로서 우리말의 글자체 가족을 바꿀 때 사용자가 직면하게 되는, 실질상 우리말의 글자체 가족을 대표하는 요소입니다.

**글자체 무게 및 폭** 작동 방식과 내용이 NFSS2와 동일합니다. 추가된 사항은 글자체 시리즈의 접두사 S로서 이는 우리말의 `\bfseries`가 `softbold`로 처리되도록 합니다.

글자체 모양 NFSS2와 같습니다.

글자체 크기 NFSS2와 같습니다.

#### Λ를 쓸 때

라텍을 쓸 때와 다른 점은 다음과 같습니다.

글자체 부호화 우리말의 부호화는 UHC로 설정되며 \fontencoding은 없습니다.

글자체 가족 모든 명령 \kscfamily나 \hfontfamily는 없고 영문과 같이 \fontfamily가 사용됩니다.

### 3.3.3 우리말 글자체의 가족 선택

#### 라텍을 쓸 때

KS X 1001의 부호 체계에 의한 상징 문자/한글/한자의 가족 구분은 글자의 부호값으로 정해지므로 특별한 명령/선언 모든 명령을 필요로 하지 않습니다. 단지 미래에 추가될 수도 있는 가능성을 위해 혹은 운영 체제 관리자를 위해, 설치된 한글라텍의 특성에 따라 다음의 모든 명령으로 상징 문자/한글/한자를 구분 설정할 수 있도록 하였습니다.

```
\kscfamily{상징문자}{한글}{한자}
```

이는 각각

```
\symboldefault
```

```
\hanguldefault
```

```
\hanjadefault
```

로 설정되어 있으며 그 값은 각각 s, w, h입니다.

우리말 문서에서의 실질적인 『우리말 글자체 가족』은

```
\textmj \mjfamily
```

```
\textgt \gtfamily
```

```
\texttz \tzfamily
```

로 선택되며 이들은 각각

```
\textrm \rmfamily
```

```
\textsf \sffamily
```

```
\texttt \ttfamily
```

와 같은 작용을 합니다. (왼쪽은 명령형, 오른쪽은 선언형)

위와 같은 모든 명령들은 우리말 가족을 바꾸어 주는 모든 명령 \hfontfamily를 사용합니다.

```
\hfontfamily{우리말가족}
```

우 리 말 가 족 에는 다음과 같은 글자체가 있습니다.

기본가족:		
mj (명조)	gt (고딕)	tz (타자)
추가가족:		
gr (그래픽)	gs (궁서)	pg (필기)
yt (옛글)	bm (봄글씨)	pn (펜글씨)
ph (펜힐림)	sh (신문)	vd (바다)
jmj (자모명조)	jgt (자모고딕)	jnv (자모노벨)
jsr (자모소라)		
dn (디나루)	pga (필기a)	

위와 같은 우리말 가족들은 상징 문자/한글/한자의 세 가족을 대표하는 외형 가족으로서, 문서에서 우 리 말 가 족 을 tz으로 지정하였을 경우 한글라텍은 한글/상징 문자/한자의 순으로 각각

wtt, wtt, wtt

의 내형 가족을 쓰도록 되어 있습니다. 외형 가족과 내형 가족의 연관은 사용자가 다음의 명령으로 문서의 어디에서든지 변경할 수 있습니다.

```
\MapHangulFamily{외형 가족}{한글 가족, 상징문자 가족, 한자 가족}
```

tz의 경우 내형 가족과의 연관은 다음과 같은 명령으로 이루어집니다.

```
\MapHangulFamily{tz}{wtt,wtt,wtt}
```

#### Λ를 쓸 때

Λ를 쓸 때는 한 글자체에 상징문자와 한글 및 한자가 모두 들어 있습니다. 그래서 라텍을 쓸 때와는 달리 상징문자/한글/한자를 구분하는 모든 명령은 필요없습니다.

다음과 같은 명령은 라텍을 사용할 때에만 사용되는 명령이며 Λ를 사용할 때에는 없는 명령입니다.

```
\kscfamily{상징문자}{한글}{한자}
\symboldefault
\hanguldefault
\hanjadefault
\MapHangulFamily{외형 가족}{내형 가족}
```

글자체의 가족은 \fontfamily로 바꿀 수 있으며 사용할 수 있는 글자체 가족은 라텍의 경우와 같습니다.

라텍의 경우와 같이 기본 글자체 이외의 글자체에 해당하는 \text??와 \??family 모든 명령의 사용을 억제하고 3.3.7의 우리말 모든 명령을 사용하도록 되었습니다.

### 3.3.4 우리말 글자체 시리즈

#### 라텍을 쓸 때

문서의 글자체 시리즈의 선택에는 NFSS2가 그대로 적용되며 라텍을 쓸 때는 한글라텍 버전 0.96까지 사용되었던 softbold를 위해 글자체 시리즈의 접두사 S가 추가되어 보통 두께의 글자체를(mdseries) 세 번 중첩 인쇄하는 기능을 합니다. 그러므로 라텍 문서에서 다음의 명령은

```
\renewcommand{\bfdefault}{Sbx}
```

혹은

```
\fontseries{Sbx}
```

우리말의 글자체 두께를 `softbold`로 처리하도록 하며 영문 글자체의 두께는 `bx`가 되도록 합니다. `hangul.sty`의 추가 선택 `softbold`는 위와 같은 기능을 애초값으로 설정합니다.

#### Λ를 쓸 때

`softbold`의 기능이 없고 두꺼운 글자체가 사용됩니다. 이 UHC 글자체를 쓰면 글자체 시리즈를 구성하는 두 요소 (넓이와 무게)가 구분되며, 넓이는 다음의 네가지 종류로 구분되고

`c` (가로 축소 75%)

`m` (보통 넓이 100%)

`x` (가로 확대 125%)

`ux` (가로 확대 200%)

무게는 `m` (보통 무게)와 `b` (진한 무게)의 두 종류가 있습니다. 예를 들어 가로로 두 배 확대 한 두꺼운 글자체를 쓸 때는 다음과 같은 명령을 줍니다.

```
\fontseries{bux}\selectfont
```

위에서, 글자체의 선택 뒤에서 `\selectfont` 명령을 줌으로써 글자체 선택이 효력을 발휘하게 됩니다.

### 3.3.5 우리말 글자체의 모양과 크기

우리말 글자체의 모양/크기는 NFSS2와 같습니다.

### 3.3.6 우리말 글자체의 애초값

문서의 애초 글자체는 NFSS2에서와 같이 다음의 명령/선언형으로 선택됩니다.

```
\textnormal \normalfont
```

한글라텍에서 애초값은 다음과 같이 설정되어 있습니다.

	한글 라텍	한글 Λ
<code>\hencodingdefault</code>	<code>H</code>	
<code>\encodingdefault</code>		<code>UHC</code>
<code>\symboldefault</code>	<code>s</code>	
<code>\hanguldefault</code>	<code>w</code>	
<code>\hanjadefault</code>	<code>h</code>	
<code>\mjdefault</code>	<code>mj</code>	<code>mj</code>
<code>\gtdefault</code>	<code>gt</code>	<code>gt</code>
<code>\tzdefault</code>	<code>tt</code>	<code>tz</code>
<code>\hfamilydefault</code>	<code>\mjdefault</code>	
<code>\familydefault</code>		<code>\mjdefault</code>

**ΛTeX** 위와 같은 애초값들은 운영 체제 관리자가 `hfont.cfg`에서 변경 설정할 수 있고, 사용자도 문서의 서두에서 재정의할 수 있습니다.

**Λ** Λ에서는 일반적으로 라텍을 사용할 때에 하는 방법으로 재정의합니다.

### 3.3.7 그외 우리말 글자체의 모듬 명령

한글라텍 버전 0.96까지 써 오던 두 글자로 구성되는 글자체 바꿈 명령은 기본 글자체 이외에 는 모두 삭제되었습니다. 이들은 라텍 버전 209 식의 글자체 바꿈 명령에 해당하며 사용을 회피 하는 것이 좋습니다.

```
\mj
\gt
\tz
```

그 반면에 우리말로 구성되는 글자체 바꿈 명령은 글자체의 가족을 선택하는 선언형으로서 계속 존재합니다.

	한글 라텍	한글 A
\명조	\textmj \mjfamily	\mjfamily
\그딕	\textgt \gtfamily	\gtfamily
\탁자	\texttz \tzfamily	\tzfamily
\그렉픽	\hfontfamily{gr}	\fontfamily{gr}
\궁서	\hfontfamily{gs}	\fontfamily{gs}
\신문	\hfontfamily{sh}	\fontfamily{sh}
\필기	\hfontfamily{pg}	\fontfamily{pg}
\펜글씨	\hfontfamily{pn}	\fontfamily{pn}
\펜흘림	\hfontfamily{ph}	\fontfamily{ph}
\봄글씨	\hfontfamily{bm}	\fontfamily{bm}
\옛글	\hfontfamily{yt}	\fontfamily{yt}
\자모명조	\hfontfamily{jmj}	\fontfamily{jmj}
\자모그딕	\hfontfamily{jgt}	\fontfamily{jgt}
\자모노벨	\hfontfamily{jnv}	\fontfamily{jnv}
\자모소락	\hfontfamily{jsr}	\fontfamily{jsr}
\박다	\hfontfamily{vd}	\fontfamily{vd}
\디나투	\hfontfamily{dn}	\fontfamily{dn}
\필기a	\hfontfamily{pga}	\fontfamily{pga}

\hfontfamily나 \fontfamily로 글자체를 지정한 후에는 \selectfont로 글자체를 선택해야 글자체가 바뀝니다.

#### 라텍을 쓸 때

\usefont는 NFSS2의 모듬 명령으로서 글자체의 부호화와 가족, 시리즈 및 모양을 한번에 정할 수 있도록 합니다.

```
\usefont{부호화}{가족}{시리즈}{모양}
```

이 모듬 명령에서 첫 번째 변수인 글자체 부호화가 \hfontencoding과 같을 경우에는 두 번째 변수를 \hfontfamily로 취급합니다. 그러므로 다음의 선언은

```
\usefont{H}{gs}{m}{n}
```



우리말 글자체 중 궁서체의 보통 두께, 보통 모양을 쓰도록 합니다. 이때 영문의 경우도 두께와 모양에서 우리말을 따르게 됩니다.

다음과 같은 명령은 NFSS2에 대응하는, 우리말을 위한 선언으로서 HFSS의 작동에 필수불가결한 모든 명령입니다. 이들은 `\begin{document}` 전에서만 선언할 수 있는 성질의 것인 만큼, 사용자의 입장에서는 보통의 경우 신경쓰지 않아도 됩니다.

```
\DeclareErrorHFont{부 오 획}{가족}{두께}{모양}{크기}
\DeclareHFontSubstitution{부 오 획}{가족}{두께}{모양}{크기}
```

#### △를 쓸 때

NFSS2의 모든 명령을 그대로 사용하며, 다음과 같이 선언되어 있습니다.

```
\DeclareFontEncoding{UHC}{-}{-}
\DeclareErrorFont{UHC}{mj}{m}{n}{10}
\DeclareFontSubstitution{UHC}{mj}{m}{n}
```

### 3.4 우리말 숫자 모든 명령

한글라텍에서는 우리말 숫자를 숫자 환경에서 쓸 수 있도록 우리말 숫자 모든 명령을 제공합니다. 우리말 숫자 모든 명령에는 다음과 같은 것들이 있습니다.

```
\jaso(ㄱㄴㄷㄹ) \gana(가나달락)
\ojaso(㉑㉒㉓㉔) \ogana(㉑㉒㉓㉔)
\pjaso(㉑㉒㉓㉔) \pgana(㉑㉒㉓㉔)
\onum(①②③④) \pnum((1)(2)(3)(4))
\oeng(㉑㉒㉓㉔) \peng((a)(b)(c)(d))
\hnum(하나둘셋) \Hnum(첫째둘째)
```

예를들어, 다음과 같이 정의를 하면

```
\renewcommand{\labelenumi}{\gana{enumi}}
\renewcommand{\labelenumii}{\jaso{enumii}.}
\renewcommand{\labelitemi}{☞}
```

다음의 원전(text)은 이렇게 나타납니다.

<code>\begin{enumerate}\디 나루</code>	
<code>\item 한글</code>	
<code>\begin{enumerate}</code>	가) 한글
<code>\item 현대어</code>	ㄱ. 현대어
<code>\item 고어</code>	ㄴ. 고어
<code>\end{enumerate}</code>	
<code>\item 漢字</code>	나) 漢字
<code>\begin{enumerate}</code>	
<code>\item 韓國語</code>	ㄱ. 韓國語
<code>\item 中國語</code>	ㄴ. 中國語
<code>\item 日本語</code>	ㄷ. 日本語
<code>\end{enumerate}</code>	
<code>\item 상징 기호</code>	다) 상징 기호
<code>\begin{itemize}</code>	☞ Ⓜ
<code>\item Ⓜ</code>	☞ W
<code>\item ㄹ</code>	☞ ㄹ
<code>\end{itemize}</code>	
<code>\end{enumerate}</code>	

### 3.5 문장부호

우리말 문서에서 사용되는 문장 부호는 라텍스에서 제공되는 문장 부호가 사용될 수 있습니다. 그러나 특수한 경우에는 우리말 문자 체계에서 제공하는 상징 문자를 사용해야 하고 경우에 따라서는 특별한 명령으로 구현되어야 합니다. [한글연구회92]의 부록에서는 우리말 문서에서 다음과 같은 문장 부호의 사용법을 규정하고 있습니다.

#### 3.5.1 마침표

- 온점(.), 고리점(°)
 

가로쓰기에는 온점, 세로쓰기에는 고리점을 쓴다.

온점은 마침표라고도 하며 영문 ASCII에서 사용되는 부호와 동일한 부호(.)로 입력되고 고리점은 우리말 상징 문자(°)가 사용된다.

젊은이는 나라의 기둥이다.
- 물음표(?)
 

의심이나 물음을 나타낸다.

영문 ASCII에서 사용되는 부호와 동일한 부호(?)이다.

이제 가면 언제 돌아오니?
- 느낌표(!)
 

감탄이나 놀람, 부르짖음, 명령 등 강한 느낌을 나타낸다.

영문자에서 사용되는 부호(!)와 동일하다.

아, 달이 밝구나!

### 3.5.2 삽표

- 반점(.), 모점( ` )  
가로쓰기에는 반점, 세로쓰기에는 모점을 쓴다. 문장 안에서 짧은 휴지를 나타낸다.  
반점은 영문자에서 사용되는 부호(.)와 동일하고 모점은 우리말 상징 부호( ` )를 사용한다.  
근면, 검소, 협동은 우리 겨레의 미덕이다.
- 가운데점(·)  
열거된 여러 단위가 대등하거나 밀접한 관계임을 나타낸다.  
라텍의 수학 기호( $\cdot$ )을 입력하거나 우리말 상징 부호(·)을 사용할 수 있다.  
철이·영이, 영수·순이가 서로 짝이 되어 웃놀이를 하였다.
- 쌍점(:)  
내포되는 종류를 들 적에, 소표제 뒤에, 저자명과 저서명 사이 등에 쓴다.  
영문자 부호 콜론(:)을 입력한다.  
문방사우: 붓, 먹, 벼루, 종이
- 빗금 (/)  
대응, 대립되거나 대등한 것을 함께 보이는 단어와 구, 절 사이에 쓴다. 분수를 나타낼 때  
에 쓰기도 한다.  
영문자(/)를 입력한다.  
착한 사람/악한 사람    맞닥뜨리다/맞닥트리다

### 3.5.3 따옴표

- 큰따옴표(“”), 겹낫표(『』)  
가로쓰기에는 큰따옴표, 세로쓰기에는 겹낫표를 쓴다. 대화, 인용, 특별 어구 따위를 나타  
낸다.  
큰따옴표는 라텍의 입력 방식에 따라 작은따옴표를 두 번(“”) 입력하여 사용하고 겹낫표  
는 우리말 상징 부호(『』)를 입력하여 사용한다. (우리말 상징 부호의 겹낫표는 시계 반대  
방향으로 90° 회전된 형태이다. 아마도 가로쓰기에서 사용될 수 있도록 변화된 것으로 보  
인다.)  
“전기가 없었을 때는 어떻게 책을 보았을까?”
- 작은따옴표(‘), 낫표(「」)  
가로쓰기에는 작은따옴표, 세로쓰기에는 낫표를 쓴다. 큰따옴표는 라텍의 인용 부호(“”)를  
입력하여 사용하고 낫표는 우리말 상징 부호(「」)를 입력하여 사용한다. (우리말 상징 부호  
의 낫표는 시계 반대 방향으로 90° 회전된 형태이다. 아마도 가로쓰기에서 사용될 수 있도  
록 변화된 것으로 보인다.)  
지금 필요한 것은 ‘지식’이 아니라 ‘실천’입니다.

### 3.5.4 묶음표

- 소괄호(())  
원어, 연대, 주석, 설명 등을 넣을 적에, 빈 자리임을 나타낼 적에도 쓴다.  
영문자의 소괄호 부호(())를 입력한다.  
우리 나라의 수도는 ( )이다.
- 중괄호({})  
여러 단위를 동등하게 묶어서 보일 때에 쓴다.  
라텍의 입력 방식에 따라 역사선 문자 다음에 해당 중괄호(\{\})를 입력한다.

주격 조사 {이  
가}

- 대괄호([])  
묶음표 안의 말이 바깥 말과 음이 다를 때 혹은 묶음표 안에 또 묶음표가 있을 때에 쓴다.  
영문자 대괄호([])를 사용할 수도 있으나 우리말 상징 부호에 있는 대괄호의 꺾임 정도가 다르므로 ([]) 우리말 상징 부호를 입력하여 사용할 수 있다.  
명령에 있어서의 불확실[단호(斷乎)하지 못함]은 복중에 있어서의 불확실[모호(模糊)함]을 낳는다.

### 3.5.5 이음표

- 줄표(—)  
이미 말한 내용을 다른 말로 부연하거나 보충함을 나타낸다.  
라텍의 입력 방식에 따라 붙임표를 세번 입력하거나 우리말 상징 문자()를 입력하여 사용한다.  
어머님께 말했다가—아니, 말씀드렸다—꾸중만 들었다.
- 붙임표(-)  
사전, 논문 등에서 합성어를 나타낼 적에, 또는 접사나 어미임을 나타낼 적에 혹은 외래어와 고유어 또는 한자어가 결합되는 경우에 쓴다.  
영문자 부호(-: hyphen)을 입력하여 사용하거나 우리말 상징 부호( )을 입력하여 사용한다. 우리말 상징 부호를 사용할 때는 부호의 좌우 여백이 적당한 지를 확인하는 것이 좋다.  
겨울-나그네    빛 에너지
- 물결표(~)  
‘내지’라는 뜻에 혹은 어떤 말의 앞이나 뒤에 들어갈 말 대신 쓴다.  
라텍 명령 \textasciitilde를 입력하거나 우리말 상징 부호(~)를 입력하여 사용한다.  
아래의 왼쪽 예문에서와 같이 라텍 명령에 사용되는 물결표는 액센트 문자를 만드는 역할을 하므로 우리말의 물결표로는 부적당하게 보인다.  
9월15일~9월25일    미술~

### 3.5.6 드러냄표

‘이나’를 가로쓰기에는 글자 위에 세로쓰기에는 글자 오른쪽에 쓴다.  
문장 내용 중에서 주의가 미쳐야 할 곳이나 중요한 부분을 특별히 드러내 보일 때 쓴다.

드러낼 낱말을 `\dotemph` 혹은 `\circemph` 명령의 변수로 지정하거나 `\dotem` 혹은 `\circem`을 선언한다. 이 명령/선언은 우리말 음절에만 적용된다.

한글의 본 이름은 훈민정음이다.

중요한 것은 왜 사느냐가 아니라 어떻게 사느냐 하는 문제이다.

### 3.5.7 안드러냄표

- 숨김표 (xx, ○○)

알면서도 고의로 드러내지 않음을 나타낸다.

라텍 수학 모드의 명령(`\times`/`\bigcirc`)을 입력하거나 우리말 상징 부호(x ○)를 입력하여 사용한다.

그 말을 듣는 순간 xxx란 말이 목구멍까지 치밀었다.

- 빠짐표(□)

글자의 자리를 비워 뚫을 나타낸다.

훈민정음의 초성 중에서 아음(牙音)은 □□□의 석 자이다.

- 줄임표(……)

할 말을 줄였을 때나 말이 없음을 나타낼 때에 쓴다.

라텍의 수학 모드 명령(`\cdots`/`\ldots`)를 입력하거나 우리말 상징 부호(…)를 입력하여 사용한다. `\cdots`는 가운데점의 위치에 연속된 점 세 개를 `\ldots`는 온점의 위치에 연속된 점 세 개를 나열한다. 우리말의 줄임 표시에 연속된 점을 세 개 사용해야 하는지 여섯 개를 사용해야 하는지는 확실하지 않다. 우리말 상징 부호는 연속된 점의 수가 세 개이다.

“어디 나하고 한 번……” 하고 철수가 나섰다.

## 3.6 우리말 각주 판짜기

영문 라텍에서의 기본적인 각주 판짜기 방식을 사용하면 각주 번호가 표시되는 각주 첫 줄이 들여 쓰여집니다. 각주 번호에는 수학식에서 멱수를 나타내는 윗첨자가 사용되고 각주 번호와 각주문을 구분하는 공간이 없습니다. 이와 같은 판짜기 방식 이외에도 출판된 우리말 문서를 보면 다른 방식의 판짜기를 볼 수 있습니다.

[이종운91] 은 각주면을 본문의 왼쪽에 맞추는 방식과 각주면을 본문의 오른쪽으로 들여 넣는 방식으로 각주면 판짜기를 대분하고 각주문의 첫줄을 들여쓰기와 내어쓰기에 의해 다시 세분하는 방식으로 각주 판짜기를 분류합니다.

한글라텍의 각주 판짜기는 영문에서의 방식을 그대로 사용합니다. 그러나 사용자의 선호도에 따른 각주 판짜기 방식을 쉽게 적용할 수 있도록 `hangulfn.sty` 패키지를 따로 제공합니다. 패키지의 사용법은 다음과 같습니다.

```
\usepackage[선택사항]{hangulfn}
```

선택사항으로는 각주 번호의 판짜기 방식을 지정하는 첨자와 괄호가 있는데 각각 다음과 같이 짜여집니다.

- 첨자: <sup>1</sup>각주문

- **괄 호**: 1) 각주문

첨 자의 경우에는 각주 번호와 각주문 사이가 붙고 괄 호의 경우에는 소괄호와 각주문 사이에 구분 공간이 있습니다. 애초값은 첨 자입니다.

각주 판면의 판짜기를 지정하는 선택사항의 종류는 다음과 같습니다.

- **다 향 이 어 쓰 기**: 연이은 각주가 새 행에서 시작하지 않고 이전 행의 오른쪽에 나열되며, 새 각주의 넓이가 행의 오른쪽 경계를 넘어 설 때에 그 각주는 새 행에서 시작한다. 문서 전체가 길이가 짧은 각주로만 구성될 때에 사용할 수 있는 판짜기 방식이다.

---

12) 첫번째 각주    13) 두번째 각주    14) 세번째 각주    15) 네번째 각주    16) 다섯번째 각주  
17) 여섯번째 각주    18) 일곱번째 각주    19) 여덟번째 각주

- **단 손 이 어 쓰 기**: “다 향 이 어 쓰 기”와 같은 방식이나 한 각주의 넓이가 행의 오른쪽 경계를 넘어 설 때에 경계를 넘어서는 부분이 줄바꿈에 의해 다음 행으로 넘겨진다는 점이 다르다.

---

12) 첫번째 각주    13) 두번째 각주    14) 세번째 각주    15) 네번째 각주    16) 다섯번째 각주    17) 여섯번째  
각주    18) 일곱번째 각주    19) 여덟번째 각주

- **내 어 쓰 기**: 각각의 각주는 새 행에서 시작한다. 각주 번호가 각주문의 왼쪽으로 내어 써지고, 각주문의 왼쪽맞춤 위치가 각주문의 첫 글자의 위치이다. 이 방식이 애초값이다.

---

12) 각주면의 각주 번호와 구분 공간을 각주문으로부터 왼쪽으로 내어 쓰는 방식이다. 각주문의 가로 넓이는 본문의 가로 넓이에서 각주 번호와 구분 공간의 넓이만큼 뺀 길이이다.

- **왼 쪽 맞 춤**: 각각의 각주는 새 행에서 시작한다. 각주의 첫 줄이 왼쪽맞춤으로 짜진다.

---

12) 각주 번호와 각주문이 들어쓰거나 내어쓰기에 의해 구분되지 않고 전체가 하나로 왼쪽맞춤 되는 판짜기 방식이다. 각주 번호와 각주문의 사이에는 구분 공간이 삽입된다. 구분 공간은 첨 자의 경우 0pt이고 괄 호의 경우 공간 문자이다.

- **들 어 쓰 기**: 라텍이 제공하는 기본 각주 판짜기와 유사한 방식이다. 각주 번호는 3배각의 위치에서 오른쪽으로 정렬되며 각주문이 구분 공간을 사이에 두고 이어진다. 각주문 사이에 줄바꿈이 일어나면 새 행이 본문의 왼쪽 맞춤 위치에서 시작한다.

---

12) 첫줄 들어쓰기 방식이다. 각주 번호의 끝이 왼쪽 가장 자리에서부터 3배각의 위치이다. 연이은 각주의 각주 번호도 같은 위치에 정렬된다.

13) 이어지는 각주이다. 각주 번호가 위쪽의 각주 번호와 함께 오른쪽으로 정렬되었다.

- **들 어 왼 쪽 맞 춤**: “왼 쪽 맞 춤”과 같은 방식이다. 전체 각주면이 본문으로부터 오른쪽으로 2배각 들어 써진다.

---

12) 각주 번호와 각주문이 들어쓰거나 내어쓰기에 의해 구분되지 않고 전체가 하나로 왼쪽맞춤 되는 판짜기 방식이다. 전체 각주면이 본문의 왼쪽맞춤 위치에서 2배각 들어진다.

- **들여내어쓰기**: “들여쓰기”와 같은 방식이다. 전체 각주면이 본문으로부터 오른쪽으로 2배각 들여 써진다.

---

12) 각주면의 각주 번호와 구분 공간을 각주문으로부터 왼쪽으로 내어 쓰는 방식이다. 전체 각주면은 본문의 왼쪽에서 2배각 들여 짜진다. 각주문의 가로 넓이는 본문의 가로 넓이에서 각주 번호와 구분 공간의 넓이 그리고 2배각의 넓이를 뺀 길이이다.

- **들여괄호맞춤**: “들여내어쓰기” 방식과 같으나 각주문에서 줄바꿈에 의해 새로 시작하는 행의 왼쪽이 각주 번호에 사용되는 괄호의 끝에 정렬되는 점이 다르다. 각주 번호 양식이 “괄호”로 바뀐다. 각주 번호와 각주문의 간격은 1배각이다.

---

12) 각주면의 각주 번호를 각주문으로부터 왼쪽으로 내어 쓰는 방식이다. 각주 번호와 각주문 사이에는 1배각의 간격이 삽입된다. 전체 각주면은 본문의 왼쪽에서 2배각 들여 짜진다. 줄바꿈 후에 새 행이 각주 번호의 괄호 다음에서 시작했다.

### 3.7 조사 이형태, 매개 모음, 지정사 탈락

`\ref`나 `\pageref` 등의 교차 참조와 `\cite` 등의 문헌 인용을 사용하면 숫자나 이름표가 자동으로 생성됩니다. 이 경우 문서를 작성하고 있는 순간에는 이 숫자나 문자가 무엇인 지를 알 수 없습니다. 그러므로 그 다음에 오게 되는 주격 조사가 ‘가’가 될 지 아니면 ‘이’가 될 지를 결정할 수 없습니다. 사용되는 변이형은 자동으로 생성되는 숫자의 음가나 이름표의 마지막 음절이 무슨 받침을 갖고 있는가에 따라 다릅니다. 매개 모음 “으”의 삽입이나 지정사 “이”의 탈락도 같은 경우입니다. 한글라텍 버전 0.96부터는 다음과 같은 음운 이형태를 자동 처리할 수 있도록 했습니다.

은/는	이/가	을/를	와/과
으로/로	으로서/로서	으로써/로써	

이를 처리해 주는 모든 명령은 각각 다음과 같습니다.

<code>\은 \는</code>	<code>\이 \가</code>	<code>\을 \를</code>	<code>\와 \과</code>
<code>\으로 \로</code>	<code>\으로서 \로서</code>	<code>\으로써 \로써</code>	

매개 모음 “으”의 삽입은 명령 “`{\_}`”를 통해 구현됩니다. 명령 이름은 “으”의 모음 “`_`”입니다. 예를 들어 “`\로`”의 정의는 “`{\_}로`”입니다. 매개 모음의 삽입은 동사의 굴절에서도 나타나지만 이를 자동으로 처리해야 할 경우는 없을 것으로 보입니다. 그 이유는 동사가 굴절을 하지만 어간이 없는 동사가 없기 때문에 매개 모음의 앞에는 정해진 동사 어간이 있기 때문입니다. 한글라텍에서 제공되지 않은 경우의 매개 모음의 처리는 “`{\_}`”를 삽입하여 해결할 수 있습니다.

지정사 “이”의 탈락은<sup>17</sup> “교수(이)시니 선생님(이)시니?” 혹은 “라텍(이)냐 워드(이)냐가 관건이다” 등과 같이 많은 경우가 있습니다. 경우의 수가 많은 이유는 지정사가 동사의 한 하위 부류로서 다양한 굴절을 보이기 때문입니다. 따라서 탈락 가능한 지정사에는 다음의 일반적인 명령을 사용합니다. 명령의 이름에 사용된 문자는 “이”의 모음 “`i`”입니다.

`{\i}`

---

<sup>17</sup> 지정사 “이다”는 학자에 따라 조사로 취급하기도 하고 상태 동사의 하나로 분류하기도 합니다. 여기서 국어학적 측면은 논외입니다.

예: `\cite{최99}{\ |}` 닥. → 참고한 문헌은 [최현배99] 닥.

`\cite` 명령으로 문헌 인용을 할 때 `plain.bst` 모양새 파일을 쓰면 인용의 이름표가 숫자로 표현되고 `halphabst` 모양새 파일을 쓰면 인용의 이름표가 문자열로 표현됩니다.<sup>18</sup> 후자의 경우, 한글라텍은 음운 변화 처리 기능을 쓰는데 있어서 문자열에 다음과 같은 제한을 두고 있습니다.

1. 문자열은 두 바이트 이상이어야 합니다. (`halphabst`를 쓰면 문자열은 보통 다섯 글자 이상이 됩니다. 우리말은 한 글자가 두 바이트 혹은 세 바이트로 구성되므로 우리말 한 글자는 문제의 대상에서 제외됩니다.)

음음: `\bibitem[HKD]{홍길동}...`  
 틀림: `\bibitem[H]{홍길동}...` ← 이름표 H는 한 바이트이므로 착오를 발생시킨다.

본문에서 `\cite{홍길동}\`을 이라 하면 “[HKD]”를 찍힙니다.

2. 문자열의 마지막 음절은 한글과 한자, 우리말 자소, 숫자 그리고 영문자로 구성됩니다. 즉, 자소를 제외한 상징 기호는 자동 음운 변화 처리에서 고려의 대상이 되지 않습니다.<sup>19</sup> 다시 말하자면, 자동 음운 변화 처리 기능은 작동하나 그 결과는 문법에 맞지 않을 수 있습니다. 이형태의 결정은 실질 형태소의 마지막 음절이 ‘ㄹ’이 아닌 중성인 경우로 간주됩니다.
3. 문자열의 마지막 글자가 영문자일 경우에, 자동 음운 변화 처리 기능은 낱말의 음가를 고려하지 않고 마지막 문자의 우리말 음가로 이형태를 결정합니다.<sup>20</sup>

음음: `\bibitem[set]{홍길동}...` ← 영문자 이름표 `set`는 마지막 문자 `t`에 의해 이형태를 결정한다. `t`의 우리말 음가는 중성으로 끝나는 음절이다. `set`의 우리말 음가는 세트로서 역시 마지막 음절이 중성으로 끝난다. 문헌 인용의 결과는 [set] 톤이다.

틀림: `\bibitem[ping]{홍길동}...` ← 영문자 이름표 `ping`은 마지막 문자 `g`가 이형태를 결정된다. `g`의 우리말 음가는 중성으로 끝나는 음절이다. 그러나 `ping`의 우리말 음가는 핑으로서 중성으로 끝나는 음절이다. 문헌 인용의 결과는 [ping] 톤이다.

`halphabst`를 사용할 경우에 인용의 부호는 저자의 이름 중 앞 세 글자와 책이 출판된 연도의 뒷 두 숫자로 구성되거나 (저자가 한 명일 때), 첫 번째 등장하는, 최대한 세 명의 저자 이름 중 첫 글자와 `\etalchar` 그리고 책이 출판된 연도의 뒷 두 숫자로 구성되므로 (저자가 여러명일 경

<sup>18</sup> `alphabst`를 쓰면 우리말이 깨집니다.

<sup>19</sup> 상징 기호가 인용의 이름표로 나올 일이 없을 것으로 생각합니다.

<sup>20</sup> 모든 영어 낱말의 우리말 음가를 올바르게 분석하기는 매우 힘듭니다.



우) 세 번째의 제한은 무시될 수 있습니다. 보통의 경우에도 `\cite`의 이름표(label)는 일련 번호이거나 (plain.bst) 저자 이름+책출판년도를 약식으로 표현합니다 (halpha.bst). 그러므로 보통은 발생하기 힘든 경우에 해당합니다. 그러나 참고 문헌 작성 시에는 이런 사실을 알고 있으므로, 발생할 가능성이 있는 오류를 사전에 방지하는 것이 좋습니다.

색인 작성 시에 `makeidx.sty`을 쓰면 `\see` 명령이 정의됩니다. 이 명령을 쓸 때는 `\see`의 변수가 빈 칸을 포함하게 되면 자동 음운 변화 처리 기능에 의해 오류가 발생합니다. 빈칸 대신에 “~”나 “\”를 쓰면 문제가 발생하지 않습니다. 예:

옳음:	<code>\index{절 see{첫장~첫절}}</code>	←	“~”나 “\”는 모두 빈 칸을 의미한다.
	<code>\index{절 see{첫장\ 첫절}}</code>		“~”는 앞뒤의 단어를 같은 줄에 축
			격하도록 하고 “\”는 다른 기능없
			이 빈 칸만 축격한다.

틀림: `\index{절|see{첫장 첫절}}`

출력: 첫장 첫절을 참고

`\pagename`, `\chaptername`과 같이 우리말화 한 한글 이름들도 자동 음운 변화 처리를 할 수 있습니다. 그러므로 ‘`\pagename\`가’라고 쓰면 “쪽이”라고 출력됩니다.

### 3.8 우리말 문헌 인용

라텍에서 문헌을 인용하기 위해서는 `\cite` 명령으로 참고 문헌 표지를 지정합니다. 참고 문헌 표지는 `thebibliography` 환경 내에서 `\bibitem`으로 지정되는 참고 문헌 항목에 주어지거나 `BIBTeX`을 이용하는 참고 문헌 데이터 파일 `.bib`의 목록에서 주어집니다. 참고 문헌 데이터 파일을 사용하면 정해진 양식으로 참고 문헌 정보를 `.bib` 파일에 수집하여 둬으로써 어느 문서에서든지 그 정보를 꺼내 쓸 수 있고 참고 문헌 양식 파일을 지정하여 다양하고 일관된 참고 문헌 목록을 작성할 수 있습니다.

참고 문헌 양식 파일은 개별 언어에 맞게 수정되어야 할 부분이 많습니다. `alpha.bst`의 경우에는 저자나 편집자와 같은 이름을 약자로 바꾸기도 하는데, 이 때에는 한글 이름에서 오류가 발생하게 됩니다. 그 이유는 한글의 한 음절이 두 바이트(EUC-KR) 혹은 세 바이트(UTF-8)로 구성되는데, 양식 파일은 이를 인식하지 못하고 영문자처럼 한 바이트만을 약자로 사용하기 때문입니다. 한글 라텍은 `alpha.bst`를 한글화한 `halpha.bst`를 제공합니다. 이 문서는 다음과 같이 `halpha.bst`를 사용하여 참고 문헌 목록을 작성하였습니다.

```
\bibliographystyle{halpha}
\bibliography{hlguide,texbook1,texbook3}
\printindex
```

위에서는 참고 문헌 양식으로 `halpha.bst`를 사용하도록 지정하고 참고 문헌 데이터 파일로 `hlguide.bib`, `texbook1.bib`, `texbook3.bib`를 지정한 것입니다.

`BIBTeX`을 이용하여 한글 참고 문헌을 만들기 위해서는 라텍이 8비트 문자를 보조 파일 (확장자가 `.aux`)에 출력할 수 있어야 합니다. 적어도 `teTeX-3.0` (`web2c-7.5.4`) 버전에서는 `tcx` 파일을 이용하여 `latex.fmt` 파일을 만듦으로써 8비트 출력이 가능하도록 되어 있습니다. 그러나 `lambda.fmt`은 `tcx` 파일을 사용하지 않을 뿐만 아니라 프로그램 원천에서도 EUC (Extended Unix Code) 문자가 `TeX`의 16진법 표기로 출력되도록 프로그래밍 되어 있습니다. 여기서 문제가 발생하는 부분은 참고 문헌 표지가 한글로 되어 있을 경우입니다. 예를 들어 참고 문헌 표지가 “홍91”일 경우에 “`\cite{홍91}`”과 같이 인용을 했을 경우에  $\Lambda$ 는 보조 파일

에 “\citation{~~ed~~99~~8d91}”을 기록하고 (UTF-8 부호화) bibtex은 이 정보를 읽어 참고 문헌 데이터 파일에서 이 표지를 갖는 참고 문헌 데이터 기재 항목을 찾아 냅니다. 그러나 참고 문헌 데이터 파일에 기록된 표지는 “홍 91”이므로 이 항목을 찾아 내지 못합니다. 따라서 한글 참고 문헌 데이터 파일을 사용하기 위해서는 보조 파일을 8비트 문자로 변환할 필요가 있고, 한글 라텍에서는 C 프로그래밍 언어로 짠 hbibtex을 제공하여 BibTeX을 사용할 수 있도록 하였습니다.

hbibtex의 사용법은 bibtex의 사용법과 동일합니다. 라텍을 사용할 경우나 참고 문헌 표지에 한글을 전혀 쓰지 않을 경우에는 hbibtex을 사용하지 않아도 됩니다.

### 3.9 우리말 색인 처리

우리말의 색인 작성은 EUC-KR 혹은 UTF-8 부호화 방식으로 만들어진 .idx 파일을 쓰는 것을 원칙으로 합니다. 이를 위해서는 latex 혹은 lambda가 8비트 문자를 보조 파일 (.aux)에 출력할 수 있어야 합니다. 소절 3.8에서 설명한 바와 같이 라텍의 경우에는 보조 파일에 8비트 문자가 출력될 수 있으나  $\Lambda$ 의 경우에는 그렇지 못합니다. 참고 문헌 인용의 경우보다 더욱 심각한 문제는 색인의 머릿문자가 바뀌는 곳이 단순히 첫 문자의 부호값이 바뀌는 곳이어서 한글 색인 생성에 적합하지 않다는 것입니다. 이를 해결하기 위해서 지금까지 한글라텍에서는 한글 색인 양식 파일과 shell/awk 스크립트를 사용하였으나 한글 라텍 버전 1.0에서는 C 프로그래밍 언어로 작성된 hmakeindex를 사용합니다.

hmakeindex은 latex이 출력하는 .idx 파일의 16진법 표기를 모두 EUC 코드로 바꾼 후에 makeindex를 실행시키고 정렬된 .ind 파일을 재정리하여 우리말을 심벌, 한글, 한자로 대분한 후 한글의 머릿글자가 바뀌는 곳에 \hindexhead 명령을 삽입합니다. \hindexhead의 애초값은 \indexspace이며 줄간격을 늘리는 역할을 합니다. 만일 색인의 머릿제목을 “『가』, 『나』, 『다』, 『라』, ...”로 표시하고자 한다면 다음과 같이 라텍 문서에서 \hindexhead를 재정의합니다.

```
\renewcommand\hindexhead[1]{\indexspace
  {\bfseries
    『 \ifcase#1 심벌\or 가\or 까\or 나\or 타\or 따\or 락\or
    마\or 바\or 빠\or 사\or 싸\or 야\or 자\or 짜\or
    차\or 카\or 타\or 따\or 야\or 한자\fi 』 }
  \nopagebreak
}
```

hmakeindex 프로그램은 C 언어로 작성되었으므로 운영 체제에서 기계어로 번역한 후 설치하여야 합니다. 윈도우즈 운영 체제에서는 미리 번역한 hmakeindex.exe를 설치하면 됩니다. 색인 작성은 다음과 같이 합니다.

```
hmakeindex <문서이름>
```

한글 색인 양식 파일은 여전히 존재하며 색인과 쪽 번호 사이를 점선으로 채우는 기능만 수행합니다. 이 모양새 파일 hind.ist를 지정하기 위해서는 hmakeindex의 추가 선택으로 -s hind를 지정합니다.

```
hmakeindex -s hind <문서이름>
```

영문자로 표기되었으나 우리말 음가로 표기되는 한글 영역에 색인을 위치시키기 위해서는 문서에서 다음과 같은 방식으로 `\index`를 지정합니다.

<pre>\LaTeX\index{라텍@\LaTeX}</pre>	←	<p><code>\LaTeX</code>은 ‘<code>\LaTeX</code>’으로 인쇄됩니다. 그러나 문자열이 ‘<code>\</code>’로 시작하기 때문에 영문자 부호 ‘<code>’</code>에 ‘<code>\LaTeX</code>’이 위치하게 됩니다. 이때는 ‘<code>@</code>’ 글자의 앞에 라텍이나 적음으로써 그 위치가 ‘라텍’이 되도록 합니다.</p>
------------------------------------	---	---

### 3.10 우리말 이름

한글라텍 0.99 버전부터는 라텍에서 사용되는 각종 이름들을 일률적으로 `\ksnamedef` 모든 명령에 의해 변경될 수 있도록 하였습니다. 예를 들어, 라텍에서 사용되는 ‘참고 문헌’의 이름 ‘Reference’는 `\refname`에 지정되어 있습니다. 이를 우리말화 하는 방법은 다음과 같습니다.

```
\ksnamedef{refname}{참고~문헌}
```

표 2에는 한글라텍에서 한글화한 “라텍 이름의 한글 및 한자화”가 나열되어 있습니다. 이 한글화는 `\begin{document}` 앞에서 위에 예를 든 것처럼 `\ksnamedef`로 재정의하면 원하는대로 변경됩니다.

`\ksTHE`는 라텍에 정의되어 있지 않고, 한글라텍에서 단원의 숫자 판짜기에 사용되도록 도입된 이름 모든 명령입니다. 이는 라텍 이름을 한글/한자화 함으로써 발생하는 문제점들을 해결하기 위해 사용됩니다. 예를 들어 `\part`의 일련 번호는 단원 숫자의 한글화로 인해 “편 I”과 같이 나오며 한글라텍에서는 `\ksTHE`를 도입하여 “제 I 편”과 같은 식으로 짜여지도록 하고 있습니다.

한글라텍에서는 `\part`와 `\chapter`, `\appendix`, `\section`까지를 위와 같이 한글화하고 있으며, 각각 다음과 같이 설정되어 있습니다.

단원 종류	숫자 앞	숫자	숫자 뒤
<code>\part</code>	<code>\ksTHE~</code>	<code>\thepart</code>	<code>~\partname</code>
<code>\chapter</code>	<code>\ksTHE~</code>	<code>\thechapter</code>	<code>~\chaptername</code>
<code>\appendix</code>	<code>\appendixname~</code>	<code>\thesection</code>	article
		<code>\thechapter</code>	book/report
<code>\section</code>	<code>\ksTHE~</code>	<code>\thesection</code>	<code>~\sectionname</code>

이를 사용자의 편의에 따라 재설정하기 위해서는 다음의 모든 명령을 사용합니다.

```
\kscntformat{단원 이름}{앞}{뒤}
```

예를 들어 `\chapter`를 “첫째 마당”과 같은 식으로 짜기 위해서는 `\begin{document}` 앞에서 다음과 같이 지정합니다.

```
\renewcommand{\thechapter}{\Hnum{chapter}}
\ksnamedef{chaptername}{마당}
\kscntformat{chapter}{}{~\chaptername}
```

라텍명령	한글라텍(한글)	한글라텍(漢字)
today	1994년 3월 6일	1994年 3月 6日
enclname	동봉물	同封物
ccname	사본	寫本
headtoname	받는이	受信人
seenname	\을 참조	\을 參考
seenname	도 참조	도 參考
contentsname	목~차	目~次
listoffigurename	그림~목차	그림~目次
listoftablename	표~목차	表~目次
refname	참고~문헌	參考~文獻
indexname	찾아보기	索~引
tablename	표	表
abstractname	요~약	要~約
bibname	참고~문헌	參考~文獻
appendixname	부록	附錄
ksTHE	제	第
partname	편	篇
chaptername	장	章
sectionname	절	節
colorlayer	환등판~색깔	幻燈版~色相
glossaryname	용어~풀이	語~彙
pagename	쪽	쪽
figurename	그림	그림

표 2: 라텍 이름의 한글 및 한자화

### 3.11 그외의 우리말화

우리말의 분철은 붙임표가 사용되지 않으므로 영어권 언어와는 다른 특별한 처리 방법이 요구됩니다.

$\LaTeX$  `hfont.tex`에서 우리말의 글자 사이가 `\hyphenchar` 없이 분리될 수 있게 하고 있습니다.  
 $\Lambda$  `ΩTP`에서 각 글자의 앞에 분철점을 집어 넣고 상황에 따라 분철 여부를 결정하도록 합니다.

영어 문서에서 단원 제목에 이은 첫 문단에서는 들여쓰기를 하지 않습니다. 그러나 한글 문서에서는 모든 문단에 들여쓰기가 적용되는 것으로 보입니다. (이는 한글 문서 작성의 표준이 규정된 것으로 보이지 않으나 한국의 출판 서적을 볼 때 그렇게 판단됩니다.) 첫 문단 들여쓰기를 하는 국가는 한국 뿐만은 아닌 것으로 보이며, 텍 운영 체제에서 `indentfirst` 꾸러미를 사용하여 이를 해결하도록 되어 있습니다. 즉, 문서의 전문 영역에서

```
\usepackage{indentfirst}
```

를 지정하면 첫 문단 들여쓰기가 쉽게 이루어집니다.

또한 들여쓰기의 정도에 있어서도 영어 문서와 우리말 문서는 차이가 있는 것으로 보입니다. (역시 정해진 표준이 있는 것으로 보이지 않고 출판사에서 관습적으로 들여쓰기의 양을 정해 쓰는 것으로 보입니다.) 한글라텍에서의 들여쓰기의 양은 한글 글자 두개의 넓이 즉 전각 2배각을 애초값으로 합니다. 이는 많은 한국어 서적에서 보이는 전각 들여쓰기보다 큰 값이지만 [이종운91]과 KS A 0104에서 채택되고 있는 값이고 우리말 문서에 적당하게 보입니다. 이 들여쓰기의 양은 다음과 같이 사용자가 원하는대로 변경될 수 있습니다.

```
\parindent15pt
```

### 3.12 PDF 문서 만들기

PDF는 Portable Document Format의 약자로서 Adobe 회사에서 제정한 문서 형식입니다. PDF 문서는 Adobe 회사의 PDF 프로그램에서 언급되는 Acrobat 문서로도 불려지고 있으며 어떤 운영 체제에서든지 미리 보기와 인쇄가 가능할 수 있는 이식성을 갖고 있습니다. PDF에는 포스트스크립트 글자체와 TrueType 글자체가 사용될 수 있고 이 문서 형식을 만들기 위해서 몇몇의 상업용 프로그램이 있습니다. 텍 사용자 층에서도 이러한 형식의 문서를 텍과 같은 강력한 식자 체계로 짤 수 있도록 많은 작업을 하였고, 그 결과로 초기의 pdf $\TeX$ 을 확장한 pdf $\TeX$ 이 web2c 버전 7.5.4에 포함되어 공개되고 있습니다. pdf $\TeX$ 에서는 포스트스크립트 글자체와 TrueType 글자체를 사용하여야 하며, KAIST 수학과와 조진환님<sup>21</sup>은 한글라텍의 포스트스크립트 글자체를 써서 한글 PDF 문서를 만들 수 있다는 소식을 알려 주신 바 있습니다.

#### 한글라텍

라텍을 사용하여 한글 PDF 문서를 만들기 위해서는 pdf $\LaTeX$ 을 사용하여야 합니다. 우리말 글자체는 포스트스크립트 글자체를 사용하므로 pdf $\LaTeX$ 을 사용하는 데 문제가 없으며 영문 글자체는 `times.sty`와 같은 꾸러미로 Adobe Type1 글자체를 사용하거나 Computer Modern 글자체를 `pfb`나 `ttf`로 형식을 변환한 글자체를 설치하여 사용해야 합니다.

우리말 문서를 다음과 같이

```
pdf $\LaTeX$  <파일 이름>.tex
```

<sup>21</sup>chofchof@math.kaist.ac.kr. 현 수원대학교 교수, chofchof@ktug.or.kr

하면 <파일 이름>.pdf 파일이 만들어집니다.

#### 한글 $\Lambda$

애석하게도 pdfomega나 pdflambda는 (아직) 없습니다. 따라서 한글 $\Lambda$ 를 사용하여 PDF 파일을 만들기 위해서는 dvi, ps, pdf의 일련의 변환을 거치거나 dvipdfm을 사용해야 합니다.

우리말 문서를 다음과 같이

```
lambda <파일 이름>.tex
dvips <파일 이름> -o
epstopdf <파일 이름>.ps
```

혹은

```
lambda <파일 이름>.tex
dvipdfm <파일 이름>
```

하면 <파일 이름>.pdf 파일이 만들어집니다.

경우에 따라서는 PDF 파일의 종이 규격이 A4 용지와 다를 경우가 있습니다. 일반적으로 pdftex를 사용할 때에 종이 규격은 pdftex.cfg에서 설정할 수 있습니다. 예를 들어 종이 규격을 국판(148mm×210mm)으로 설정하기 위해서는 다음과 같이 pdftex.cfg를 수정합니다.

```
page_width 148 true mm
page_height 210 true mm
```

수정된 pdftex.cfg 파일은 작업 중인 자료방에 두면 됩니다.

그러나 ps 파일을 거쳐서 pdf 파일을 만들 때에는 dvips로 ps 파일을 출력하는 단계에서 선택 사항으로 종이 크기를 정해 주어야 합니다.

```
dvips -t 148mmx210mm <파일 이름>.dvi -o
```

pdf 파일을 미리보기 위해서는 adobe 회사에서 공개한 acroread나 ghostscript 혹은 유닉스 운영 체제에서만 제공되는 xpdf<sup>22</sup>를 사용할 수 있습니다. 라텍으로 문서 작업을 할 때에는 원천 문서에서 수정된 내용의 결과를 미리보기를 통해 확인하는데, 유닉스 운영 체제의 xdvi나 윈도우즈 운영 체제의 yap과 같은 dvi 미리보기 프로그램은 미리보기 화면의 “재생(refresh)” 기능을 통해 이미 열려져 있는 창에서 수정된 결과물을 간단히 확인 할 수 있습니다. 그러나 pdf 미리보기 프로그램은 xpdf만이 “재생” 기능이 있어서 다른 프로그램을 사용할 경우에는 조금 불편한 점이 있습니다.

<sup>22</sup><http://www.aimnet.com/~derek/xpdf/>

## 부록 A 참고 문헌 데이터베이스

라텍으로 문서의 판짜기에 익숙해진 후에는 참고 문헌 목록의 관리에 관심을 가질 수 있습니다. 참고 문헌 데이터베이스는 학술지 출판과 같이 전문 분야의 학회에서 유용하게 사용될 수 있고, 참고 문헌 양식 파일을 학회에서 제공함으로써 참고 문헌 목록의 판짜기가 일관성을 가질 수 있게 됩니다. 또한 참고 문헌 데이터베이스는 특정 분야에 독립적인 형식으로 구축되고 이 데이터베이스를 이용하는 사용자가 필요한 정보를 원하는 모양으로 출력할 수 있습니다.

한글 참고 문헌 데이터베이스를 구축하기 위해서는 영문 참고 문헌 데이터베이스와 다른 특별한 방식이 없습니다. 다만, 주의할 점이 있다면 데이터베이스의 기록 사항 중에서 특별한 기능을 갖는 기능이 데이터베이스의 내용과 동일한 문자 세트를 사용하고 특별한 표시가 없으므로 오용될 수 있다는 점입니다.

참고 문헌 데이터베이스 파일의 확장자는 .bib이며 각 문헌이 정해진 순서 없이 다음과 같은 틀을 유지하며 나열됩니다.

```
@<문헌 유형> = {<라벨>,
  <필수 사항> = {<내용>},
  ...
  <선택 사항> = {<내용>},
  ...
}
```

<라벨>은 문서에서 문헌을 인용할 때 사용되는 참고 문헌 표지이며 <필수 사항>과 <선택 사항>은 <문헌 유형>에 따라 다릅니다.

### A.1 문헌 유형

대부분의 참고 문헌 양식 파일에서 사용되는 문헌의 유형과 필수/선택 사항은 다음과 같으며 필수/선택 사항이 아닌 사항은 bibtex에 의해 무시됩니다.

- **article**: 정기 간행물이나 잡지의 논문  
필수: author, title, journal, year  
선택: volume, number, pages, month, note
- **book**: 출판된 서적  
필수: author 혹은 editor, title, publisher, year  
선택: volume 혹은 number, series, address, edition, month, note
- **booklet**: 등록된 출판사나 기관을 통하지 않고 인쇄/제본된 책  
필수: title  
선택: author, howpublished, address, month, year, note
- **inbook**: 출판물의 장, 절 혹은 쪽 범위  
필수: author 혹은 editor, title, chapter 내지 pages, publisher, year  
선택: volume 혹은 number, series, type, address, edition, month, note
- **incollection**: 출판물을 구성하는, 자체의 제목을 가지고 있는 일부분  
필수: author, titiel, booktitle, publisher, year

선택: editor, volume 혹은 number, series, type, chapter, pages, address, edition, month, note

- **inproceedings**: 학회 회보에 발표된 논문  
필수: author, title, booktitle  
선택: editor, volume 혹은 number, series, pages, address, month, organization, publisher, note
- **manual**: 기술 문서  
필수: title  
선택: author, organization, address, edition, month, year, note
- **masterthesis**: 석사 학위 논문  
필수: author, title, school, year  
선택: type, address, month, note
- **misc**: 어느 유형에도 속하지 않을 때 (선택 사항이 하나도 없으면 오류)  
필수: 없음  
선택: author, title, howpublished, month, year, note
- **phdthesis**: 박사 학위 논문  
필수: author, title, school, year  
선택: type, address, month, note
- **proceedings**: 학회 회보  
필수: title, year  
선택: editor, volume 혹은 number, series, address, publisher, note, month, organization
- **techreport**: 교육 기관과 같은 기관에서 월간/계관과 같은 연속물로 출판되는 보고서  
필수: author, title, institution, year  
선택: type, number, address, month, note
- **unpublished**: 형식적으로 출판되지는 않았지만 저자와 제목을 가지고 있는 문서  
필수: author, title, note  
선택: month, year

## A.2 기재 사항

각 기재 사항의 내용은 다음과 같습니다.

- **address**: publisher나 기관의 주소. 넓은 지역에 분포한 출판사의 경우에는 도시 이름만 기재한다. 작은 출판사의 경우에는 전체 주소를 기재하는 것이 독자에게 유용할 것이다.
- **annotate**: 메모. 표준 참고 문헌 양식에서는 사용되지 않지만 **annotate**와 같이 주해 참고 문헌을 생성하는 양식에서 사용된다. 이 터는 새 문장으로 시작하므로 영어의 경우에는 첫 글자가 대문자이어야 한다.
- **author**: 저자의 이름이며 **BIBTEX**의 이름 형식으로 기재한다.



- **booktitle**: 일부분만 인용되는 서적의 제목. 서적 등재 사항에는 **title**이 사용된다.
- **chapter**: 장 번호.
- **crossref**: 참고 문헌 데이터베이스에 등재된 참고 문헌의 교차 참조 표지.
- **edition**: “개정판”과 같은 간행물의 발행 판수. 서수로 주어져야 하며 영어의 경우 “Second”와 같이 첫 글자가 대문자이어야 한다. 표준 양식을 통해 필요한 경우에는 첫 글자가 소문자로 바뀐다.
- **editor**: 편집자의 이름. **BIBTEX**의 이름 형식으로 기재한다. **author**와 함께 기재된 때에 **editor**는 참조가 등장하는 서적의 편집자를 나타낸다.
- **howpublished**: 통상적인 간행물이 아닌 경우 출판된 방식.
- **institution**: 기술 문서를 후원한 기관.
- **journal**: 정기 간행물의 이름. 잘 알려진 정기 간행물의 경우에는 약어를 사용할 수도 있다.
- **key**: 목록의 정렬이나 교차 참조 그리고 **author**나 **editor**가 없을 경우에 문헌 표지를 만드는데 사용된다. `\cite` 명령의 표지 및 데이터베이스 등재 항목의 처음에 나오는 문헌 표지와 혼동해서는 안된다.
- **month**: 출판물이 출판된 달이나 출판되지 않은 출판물이 쓰여진 달. 영어의 경우 일관성을 위해 세 문자 약어로 **jan**, **feb**, **mar** 등이 사용되어야 한다.
- **note**: 독자에게 유용하다고 생각되는 추가적인 정보.
- **number**: 정기 간행물이나 잡지, 기술 문서 혹은 연재물 내용의 번호. 정기 간행물이나 잡지의 판이나 호는 보통 **volume**과 **number**로 식별되고 기술 문서는 보통 **number**로 식별되며 이름을 갖는 연재물의 서적은 가끔 **number**를 갖기도 한다.
- **organization**: 학회를 후원하거나 **manual**을 출판한 기관.
- **pages**: 하나 이상의 쪽이나 쪽 범위로 42--111, 7, 41, 73--97 혹은 43+로 기재되는데 여기서 ‘+’는 단순하지 않은 범위의 쪽을 가리킨다.
- **publisher**: 출판사 이름.
- **school**: 논문이 쓰여진 학교의 이름.
- **series**: 연재물이나 여러 권으로 구성되는 출판물의 이름. 전체 출판물을 인용할 때에 **title**은 제목을 말하고 선택 사항인 **series**는 연재물의 이름이나 여러권으로 구성된 출판 서적의 이름을 말한다.
- **title**: 책의 제목.
- **type**: 기술 문서의 유형을 지정하며 “연구 노트”와 같이 기재한다. 유형이 주어지지 않으면 애초값인 “기술 문서”가 사용된다. 영어의 경우에는 애초값이 “Technical Report”이다. **phdthesis** 유형의 경우에는 “박사 학위 논문” (애초값)이라고 명시할 수 있다. 영어의 경우에는 “{Ph.D.} dissertation”이 애초값이다. **inbook**이나 **incollection** 유형

에서도 chapter = "1.2", type = "절"이라고 기재함으로써 애초값인 “1.2 장” 대신에 “1.2 절”을 얻을 수 있다.

- volume: 연속 간행물이나 여러 권으로 구성된 인쇄물의 권.
- year: 출판 연도. 출판되지 않은 서적의 경우 서적이 쓰여진 연도. 일반적으로 1984와 같이 네 개의 숫자로 구성되지만 표준 참고 문헌 양식은 “about 1984”와 같이 마지막 네 문자가 숫자인 경우를 잘 처리한다.

### A.3 저자 및 편집자 이름 형식

author나 editor의 영어 이름은 다음의 두 가지 형식으로 입력됩니다.

"<이름> <성>"

"<성>, <이름>"

그러나 한국어 이름은 공백 문자가 없이 다음과 같은 형식으로 입력됩니다.

<성><이름>

영어 이름을 반점으로 성과 구분하는 방법은 성이 여러 낱말로 구성된 경우에 효과적입니다.

이름에 사용되는 기능어에는 and와 others가 있습니다. and는 여러 이름을 나열할 때 사용되고 others는 여러 이름을 축약할 때 사용됩니다. 따라서 이름 기재 내용이 and others로 끝날 때에 이는 “\_외”(영어의 경우는 “*et al.*”)로 출력됩니다.

## 부록 B 한글화

EUC-KR 부호화 문자를 처리하는 한글라텍에서 발생하는 문제점들은 대부분이 문자의 범주에 기인합니다. 텍의 명령은 더 이상 expand 될 수 없을 때까지 expand 되는 반면에 문자는 더 이상 expand 되지 않는 범주에 속합니다. 영어 알파벳 문자는 모두 문자 범주에 속하므로 예를 들어 “word”를 구성하는 네 개의 문자는 expand 후에도 그대로 남습니다. 반면에 EUC-KR로 부호화된 “낱말”은 네 개의 부호가 두 개씩 쌍을 이루고 있으며 각 쌍이 두 개의 한글 음절 “낱”과 “말”을 형성합니다. EUC-KR 부호화 방식은 한 음절을 구성하는 두 개의 바이트 값에 의해 한국어 음절 값이 정해지므로 한글라텍에서는 첫 번째 바이트를 두 번째 바이트를 인수로 취하는 명령으로 인식하도록 하고 있습니다. 그 결과 한글 음절은 expand 된 후에 소위 말하는 원시 명령으로 구성되는 일련의 명령 집합체로 변합니다.

문자 범주를 기대하는 명령의 변수가 한글일 때에 라텍은 오류를 유발합니다. \label이나 \cite 명령 등의 변수가 한글일 때가 그렇습니다. 한글라텍에서는 표준 문서 양식에서 사용되는 이와 같은 명령들을 수정하여서 한글 라벨을 사용할 수 있도록 했습니다. 그러나 다른 패키지에 서 이런 명령들을 재정의하면 또다시 문제가 발생할 수 있습니다.

showkeys 패키지는 \label, \ref, \pageref, \cite, \bibitem 명령을 수정하여 내부적으로 사용되는 표지를 출력하는 기능을 수행합니다. 한글라텍에서는 이 패키지를 한글화 하여 showkeys.sty의 이름으로 제공합니다.

showtags 패키지는 참고 문헌의 인용에 사용되는 표지를 showkeys 패키지와는 다른 방식으로 출력하는 기능을 수행합니다. 이 패키지를 한글화 한 파일은 showtags.sty입니다.

`hyperref` 패키지는 주로 `pdflatex`과 함께 쓰여 PDF 문서 내에서의 교차 참조 사이를 항해할 수 있는 기능을 제공합니다. 한글라텍에서는 이 패키지도 한글화 하여 `khyper.sty`로 제공합니다. 다만 `bookmarks` 기능은 제대로 구현하지 못했습니다. 그 이유는 한글 문자의 부호화가 EUC-KR이기 때문이며 PDF에서는 유니코드를 요구합니다. `bookmarks`의 한글화는 EUC-KR 부호화 문자를 출력하는 수준에서 그쳤습니다. 따라서 `pdflatex`의 `bookmarks` 기능은 사용하지 않는 편이 좋습니다.

한글 $\Lambda$ 에서는 UTF-8로 부호화된 한글이 16비트의 문자 범주로 처리됩니다. 따라서 한글라텍의 경우에는 달리 `expand`에 의한 오류가 발생하지 않습니다. 그러므로 `showkeys`, `showtags`, `hyperref`에서 요구하는 한글화 작업이 필요하지 않습니다. 만일 한글 $\Lambda$ 에서 위의 한글화된 패키지를 사용했을 때는 한글화 되지 않은 기존 패키지만 사용되고 한글화 부분은 적용되지 않습니다.

## 참고 문헌

- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, Reading, MA, USA, 1994.
- [KD99] Helmut Kopka and Patrick W. Daly. *A Guide to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>: Document Preparation for Beginners and Advanced Users*. Addison-Wesley, Reading, MA, USA, third edition, 1999.
- [Knu86a] Donald E. Knuth. *The T<sub>E</sub>Xbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [Knu86b] Donald E. Knuth. *T<sub>E</sub>X: The Program*, volume B of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [Knu86c] Donald E. Knuth. *The METAFONTbook*, volume C of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [Knu86d] Donald E. Knuth. *METAFONT: The Program*, volume D of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [Kop91] Helmut Kopka. *L<sup>A</sup>T<sub>E</sub>X—Eine Einführung*. Addison-Wesley Verlag, Bonn, Germany, third edition, 1991.
- [Lam85] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X—A Document Preparation System—User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.
- [Sch94] Peter Scherber. *Offizin—Schriftenreihe zu T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, und Metafont (Band 1, Hrsg. von DANTE e. V)*. Addison-Wesley, Reading, MA, USA, 1994.
- [이종운91] 이종운. 「도서편집총람」. 범우사, 서울, 초판, 1991.
- [최우형92] 최우형. hlatex 가이드. 기술 문서, KAIST, 1992. available from uhyung@daiduk.kaist.ac.kr.
- [최현배99] 최현배. 「우리말본」. 정음문화사, 서울, 18판, 1999.
- [한글연구회92] 한글연구회. 「새 한글 맞춤법 및 용례집」. 이사야 법조각, 서울, 초판, 1992.
- [홍이김<sup>+</sup>91] 홍운표, 이기문, 김충희, 김홍규, 박성중, 정인상, 김병선, 김홍배, 및 이민재. 한글 옛글자의 컴퓨터 처리방안에 대한 연구. 「한국어전산학」, 창간호, 8 1991.

## 찾아보기

- dvipdfm, 34
- dvipdfm, 34
- alpha.bst, 28
- hbibtex, 7, 30
- hfont.sty, 12
- HFSS, 14, 16
- hglo.ist, 7
- hind.ist, 7
- \hindexhead, 30
- hmakeindex, 7, 30
- \ksnamedef, 31
- L<sup>A</sup>T<sub>E</sub>X3 개발, 2
- \MapHangulFamily, 18
- NFSS2, 14
- PDF 문서, 33
- pdflatex, 33
  - 종이 규격, 34
- pdftex.cfg, 34
- softbold, 18
- TDS, 3
- texmf.cnf, 4
- web2c-7.0, 2
  
- 『심벌』
  - 각주 판짜기, 25
  
- 『가』
  - 글자체
    - 구성 요소, 14
    - 넓이, 8
  
- 『나』
  - 내형 글자체 바꿈 명령, 15
  
- 『라』
  - 라텍, 1
  - 개발 목표, 2
  - 글자체 선택, 14
  - 버전 3, 2
- Λ, 3
  
- 『마』
  - 매개 모음, 27
  - 명령형, 14
  - 문서 식자 시스템
    - 택, 1
  - 문서의 모양새, 12
  - 문헌 인용, 29
  
- 『사』
  - 색인, 30
    - \hindexhead, 30
  - 선언형, 14
  
- 『아』
  - 역사
    - L<sup>A</sup>T<sub>E</sub>X, 1
    - L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, 2
    - T<sub>E</sub>X, 1
    - hlatex, 2
    - 한글라텍, 2
- Ω, 2, 3
- 외형 가족, 18
- 외형 글자체 바꿈 명령, 15
- 우리말 글자체 가족, 17
- 우리말 이름, 31
  
- 『자』
  - 자동 음운 변화, 27
    - 색인, 29
  - 자동 음운 변화 처리, 28
  - 자동 조사
    - 인용, 28
  - 조사 이형태, 27
  - 출간적, 13
  - 지정사 탈락, 27
  
- 『차』
  - 추가 선택 처리 방식, 11

『과』

판형, 11

『하』

한글라텍, 2

글자체, 8

글자체 바꿈 명령, 20

글자체 선택, 13

글자체 종류, 18

단원 관짜기, 31

문서의 틀, 11

분칠, 33

사용법, 11

설치, 3

.fd, 7

fontmap, 8

halpha.bst, 7

hangul.sty, 6

hlatex.ocp, 7

글자체, 7

라텍, 9

모듬, 5

숫자 모듬 명령, 21

추가 선택, 11

환경형, 14