

한국어 텍 ko.T_EX v0.1.0 사용 설명서

은광희 · 김도현 · 김강수

2007 년 7 월

일러두기

ko_ℒTeX은 H_ℒTeX과 hangul-ucs가 결합하여 탄생한, 명실 상부한 “한국어(한글) 텍 시스템”입니다. “한글 라텍(H_ℒTeX)”은 1990년대 이후 텍에서의 한글 사용에 사실상 표준이었으며, hangul-ucs는 한글 라텍의 현저한 영향 아래서 성장한 유니코드 한글 텍 매크로였던 것입니다.

이제 한국텍학회(KTS)의 주도 아래, 한글 라텍과 hangul-ucs가 하나의 매크로 시스템으로 통합된 것을 매우 기쁘게 생각합니다. 이와 더불어 더욱 알차고 유용한 사용자 안내서를 작성하기 위해 많은 노력을 하였습니다.

한글 라텍의 표준 문서였던 “한글 라텍 길잡이”(은광희)는 아마도 가장 많이 읽힌 한글로 된 라텍 관련 문서 중 하나가 아닐까 합니다. 이 문서는 단순히 한글 사용법을 넘어서서, 텍에 입문하는 데 있어서도 많은 도움을 주었습니다.

이 문서는 한글 라텍 길잡이를 바탕으로 ko_ℒTeX에 맞도록 개정하여 쓴 것입니다. 감사의 말과 머리말은 거의 그대로 보존하였으며, ko_ℒTeX이 등장하는 경과를 설명하는 소절을 하나 추가하는 정도에 그쳤습니다. 다만 문체의 일치를 유지하기 위하여 이 일러두기와 감사의 말을 제외하고는 경어체를 평어체로 고쳐 썼습니다. 또한 사용법에 있어서 한글 라텍 이후의 여러 가지 발전 사항을 새로운 편제로 서술하려 하였습니다. 지나치게 기술적인 서술을 피하고 실제 사용자에게 구체적인 지침을 제공하는 데 이 문서의 목적이 있습니다.

이 문서가 ko_ℒTeX을 사용하는 분에게 조금이라도 도움이 되기를 바랍니다. 뜻하지 않은 잘못된 내용이나 실수, 빠진 내용, 오식을 바로잡아 주시면 더 나은 사용자 안내서를 만드는 데 크게 도움이 될 것입니다.

은광희 김도현 김강수 (識)

감사의 말

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

이 안내서에서 다음과 같은 정보를 얻을 수 있습니다.

- 한글 라텍 역사.
- 한글 라텍 설치.
- 한글 라텍 사용.
- 우리말 글꼴 선택.

이 글이 한글 라텍을 사용하는 분들에게 조그마하나마 도움이 될 수 있기를 바랍니다.

한글 라텍을 발표하면서 한글 글자체 작성에 도움을 주신 고려대학교 언어학과 李基用 교수님, 문화관광부¹ 포스트스크립트 글자체를 제공하신 연세대학교 국문과 洪允杓 교수님, 한글을 보는 눈을 깨우쳐 주신 충남대학교의 노용균교수님, 한글 라텍의 문제점과 개선점을 지적하고 도와주신 원세연님과 이천우님, 우리말 글자체 선택 모듬 명령(macro)을 제안·수정·실험해 주신 이형석님, 항상 컴퓨터에서의 한글 사용에 대해 전반적으로 도움을 주시는 신정식님, KTUG을 운영하시는 조진환님, 한글 글자체를 비롯하여 다방면에서 활동하시는 박원규님, 한글 라텍 윈도우즈 포팅에 노력을 아끼지 않으시는 김강수님 그리고 한글 라텍의 발전에 기여하신 많은 여러분께 감사의 마음을 전합니다. 특히 李基用 교수님은 한글 라텍의 발전을 위하여 2GB 굳은 저장판(hard disk)을 2개 기증하셨습니다. 한글 라텍은 이를 바탕으로 새로운 완성형 글자체와 UHC 글자체들을 만들 수 있었습니다. 한글 라텍을 완성해 가면서 李基用 교수님께 감사합니다.

— 殷 光熙.

¹이전의 문화체육부

차 례

제 I 편	총론	8
제 1 장	텍, 라텍, 한글 라텍	9
제 1 절	텍에 대해	9
제 2 절	라텍에 대해	9
제 3 절	한글 라텍에 대해	10
제 4 절	Omega(Ω)에 대해	12
제 5 절	유니코드를 이용한 실험들 (2002–2006)	12
제 6 절	ko \TeX 의 탄생 (2007)	14
제 2 장	설치	16
제 1 절	\TeX 배포판	16
제 2 절	$\epsilon\text{-}\TeX$	17
제 3 절	디렉터리 구조 (TDS)	17
제 4 절	일반적인 설치 지침	19
4.1	다운로드	19
4.2	매크로와 유틸리티	19
4.3	글자체의 설치	20
제 5 절	각 플랫폼별 설치 패키지	21
제 6 절	CVS 개발 버전	21
제 3 장	사용	22
제 1 절	기본 사용법	23
제 2 절	pdf 문서 작성을 위한 설정	24
2.1	ko \TeX /utf 문서	24
2.2	ko \TeX /euc 문서	25
2.3	요약	26
제 3 절	폰트에 대하여	27
3.1	기본 글꼴	27
3.2	추가 글꼴	28

3.3	은 글꼴 트루타입 폰트 세트	29
3.4	타자 글꼴에 관한 주석	29
3.5	요약	29
3.6	폰트의 라이선스	30
제 4 절	utf 버전과 euc 버전의 관계	30
제 II 편 ko.TeX/utf		32
제 4 장 유니코드 문서 작성하기		33
제 1 절	개관	33
제 2 절	첫번째 문서	33
2.1	간단한 문서 작성	33
2.2	컴파일	34
2.3	영문 글꼴	34
2.4	옵션들	35
2.5	부가 패키지들	35
제 3 절	글꼴 사용과 선택	36
3.1	기본 글꼴을 변경하는 방법	37
3.2	일시적으로 폰트를 변경하는 방법	39
3.3	사용자 폰트를 활용하는 방법	40
3.4	엄격한 문자 체크	41
제 4 절	몇 가지 한글화	41
4.1	기호문자 처리	41
4.2	한글 문서 서식	43
4.3	장절명령	43
4.4	강조	43
4.5	자간, 행간	46
4.6	장평	46
4.7	nonfrench spacing	47
4.8	각주	48
4.9	한글식 카운터	48
4.10	자동조사 명령	49
4.11	enumerate의 항목 머리	50
제 5 절	미세 조정	52

5.1	수식과의 간격	52
5.2	한글 폰트스팩	52
5.3	Micro Typography	54
제 6 절	행나눔	56
6.1	영문자와 숫자 뒤의 한글 개행	56
6.2	여는 괄호 앞의 행자름	56
6.3	이른바 “고아” 문제	57
제 7 절	한글 PDF 만들기	58
7.1	하이퍼링크	58
7.2	폰트 문제	59
7.3	pdf 한글 텍스트의 검색과 추출	59
7.4	한글 책갈피	60
제 8 절	찾아보기와 참고문헌	60
제 9 절	옛한글 구현	61
제 10 절	다른 패키지 및 클래스와의 호환성	62
10.1	$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ 클래스	62
10.2	beamer 지원	62
10.3	prosper/powerdot 지원	63
제 11 절	일본어와 중국어 문단	63
제 5 장	memhangul과 oblioir	66
제 1 절	소개	66
제 2 절	memhangul	66
제 3 절	oblioir	67
3.1	간단한 문서 만들기	67
3.2	용지, 여백, trim mark	68
3.3	글꼴 선택	68
3.4	기타	68
제 III 편	ko$\mathcal{T}\mathcal{E}\mathcal{X}$/euc	69
제 6 장	ko$\mathcal{T}\mathcal{E}\mathcal{X}$/euc의 의의와 사용법	70
제 1 절	ko $\mathcal{T}\mathcal{E}\mathcal{X}$ /euc의 의의	70
1.1	문자의 범위	70

1.2	ko _{TeX} /euc의 한계	70
1.3	하위 호환성	71
1.4	사용설명서에 관하여	71
제 2 절	ko _{TeX} /euc의 사용 설명	71
2.1	한글 문서의 틀	71
2.2	문서의 모양새 설정	72
2.3	글자체 선택	74
2.4	우리말 숫자 모듬 명령	83
2.5	우리말 각주 판짜기	84
2.6	조사 이형태, 매개 모음, 지정사 탈락	85
2.7	우리말 이름	88
2.8	그외의 우리말화	89
제 3 절	한글화	90
제 IV 편 한글 문서 작성		92
제 7 장 한글 문서 서식		93
제 1 절	각주 형식	93
1.1	hangulfn의 각주 판짜기	93
1.2	각주 문단 모양 관련	95
제 2 절	자간, 행간, 단어간격	96
2.1	자간	96
2.2	단어간격	97
2.3	행간	98
제 3 절	ko _{TeX} /euc에서의 밑줄	99
제 4 절	장절표제	100
제 5 절	문장부호	100
5.1	우리말 문장부호의 사용법	100
5.2	문장부호에 관련된 문제	104
제 8 장 색인과 문헌목록		105
제 1 절	ko _{TeX} /euc의 h _{bib} t _{ex} 과 h _{alpha}	105
1.1	우리말 문헌 인용	105
1.2	참고 문헌 데이터베이스	106

제 2 절 색인	111
2.1 ko \TeX /utf의 색인 처리	111
2.2 ko \TeX /euc의 우리말 색인 처리	111
제 9 장 한글 트루타입 폰트의 사용	113
제 1 절 ttf2kotexfont 사용법	113
제 2 절 문서에서 폰트 설정하기	114
제 3 절 pdf 문서 제작을 위한 pdf \TeX 사용	115
제 10 장 plain\TeX과 ko\TeX	116
제 1 절 kotexplain.tex	116
제 2 절 hangulcweb.tex	117
제 11 장 html 제작	118
참고 문헌	119
부록	124
부록 A 완성형 한글 문자	124
부록 B 윈도우즈 유니코드 편집기	127
부록 C 이 문서에 대하여	129
부록 D 글꼴 예문	130
부록 E 용어 대조	134

제 I 편

총론

제 1 장 텍, 라텍, 한글 라텍

제 1 절 텍에 대해

텍(TeX)은¹ 미국 스탠퍼드 대학 전산학과의 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 사이에 Knuth 교수가 다시 완성시킨 문서 식자 체계이다. 텍은 원래 SAIL² 언어로 쓰여졌으나 1979년 초에는 Knuth 교수와 Luis Trapp Pardo에 의해 공동 개발된 WEB 언어로 전환시키는 작업이 시작되었고 1979년과 1980년에 Ignacio A. Zabala 씨가 이를 완성시켰다. 텍은 1979년 후반기와 1980년 전반기에 걸쳐 Knuth 교수가 더욱 안정된 프로그램으로 정착시켰고 그해 9월에 TeX82 버전 0으로 발표되기 시작했다. 그 후 많은 사람들의 조언을 반영하여 1989년 9월에 최종 검토를 거쳐 발표된 텍은 현재, 전문적인 자연 과학 문서를 작성하는 데 가장 좋은 문서 식자기로 인정받고 있다((Knuth, 1984) – (Knuth, 1986)을 참고). 수학자이기도 한 Knuth 교수는 이 문서 식자 체계를 사용하여 수학 공식도 자유 자재로 쓸 수 있도록 했다. Knuth 교수는 텍 외에도 METAFONT를 개발하여 텍 문서를 인쇄하는 데 필요한 양질의 글자체를 만들어 낼 수 있도록 하였고 이로써 텍은 출판 서적을 만드는 데 손색이 없는 문서 식자 체계로 자리를 잡게 되었다.

제 2 절 라텍에 대해

텍의 식자 기능은 매우 탁월하지만 일반 사용자의 측면에서는 프로그래머의 사고

¹‘텍’은 그리스어 $\tau\epsilon\chi$ 에서 유래된 낱말이다.

English words like ‘technology’ stem from a Greek root beginning with the letters $\tau\epsilon\chi$...; 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 χ 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.

²Stanford Artificial Intelligence Language

능력을 요구하기 때문에 많은 노력과 경험으로 텍의 기능을 이해하여야 문서를 작성할 수 있다. 예를 들어 사용자는 글자체를 스스로 정의하고 문장과 문장 사이의 간격을 넓혀주는 등의 일을 일일이 손으로 처리해야 했다. 그래서 텍은 전문적인 프로그래머 층에서나 사용할 수 있었다.

그러나 미국의 전산과학자인 Leslie Lamport는 사용자와 텍의 사이를 좁히는 역할을 할 수 있도록 라텍(L^AT_EX)을 개발하여 쉽게 문서를 작성하게끔 기여하였다 ((Lamport, 1985)를 참고). 그는 일반적인 문서 모양새를 텍의 모듬 명령(macro)으로 정의함으로써, 프로그래밍의 경험이 없는 사용자도 쉽게 문서를 작성할 수 있도록 하였다. 사용자는 쓰고자 하는 글의 전체적인 윤곽에 맞는 양식을 지정한 후, 작성하는 문서의 내용에만 집중하면 원하는 모양새의 문서를 작성할 수 있게 되었다. 그리하여 사용자는 텍에 대한 전문 지식이 없어도 짧은 시간 내에 좋은 문서를 작성할 수 있게 되었다.

세월이 흐르면서 라텍은 전세계적으로 많이 보급되었고 이에 따라 라텍이 해내야 하는 과제의 폭도 넓어졌다. 그 결과로 라텍은 근본적인 바탕을 바꾸고 국제화되어야 한다는 여론이 높아졌고 1992 말부터는 라텍 버전 3 일감 팀(project team)³이 구성되어 새로운 라텍 버전 3의 개발 작업에 착수하였다. 1994년 6월 1일을 기해 라텍 버전 2 ϵ 을 공식적으로 내놓은 이 팀은 현재 (94년 11월) 깃치 수준(patch-level) 4를 발표하면서 계속 라텍 버전 3을 위해 작업하고 있다. Frank Mittelbach 씨와 Rainer Schöpf 씨는 (Scherber, 1994)에서 라텍 버전 3 개발의 목표를 다음과 같이 설명하고 있다.

- 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.

제 3 절 한글 라텍에 대해

이렇게 라텍이 국제화되어 가는 동안 한국 KAIST 전산학과에서는 라텍으로 한글을 쓸 수 있도록 모듬 명령을 만들고 한글 글자체를 준비하여 hlatex을 (최우형,

³Leslie Lamport, Johannes Braams, David Carlisle, Alan Jeffrey, Frank Mittelbach, Chris Rowley, Rainer Schöpf: 이들은 (Mittelbach et al., 2004)를 출간하였는데 이 책의 수입은 그 절반이 라텍 버전 3 개발팀에게 할당되어 이 팀을 지원하게끔 되어 있다.

백운주) 보급하였다.⁴ ((최우형, 1992)를 참고) `hlatex`은 기존의 라텍에 한글 글자체를 추가하는 `hfont.tex`, 한글 문서 작성 환경을 정의하는 `harticle.sty`과 `hreport.sty`, `hbook.sty` 그리고 한글 KS 완성형 부호 체계를 텍의 모든 명령으로 변환해 주는 앞처리기(preprocessor) `htex`으로 구성되어 있었다. 보급되는 글자체의 사용을 제한받으면서 출발한 `hlatex`은 그런대로 통신망을 통해 KAIST 외부의 일반에게 퍼지게 되었고⁵, 이를 바탕으로, 한자의 사용을 추가한 `jhtex`이 (殷光熙) 만들어지게 되었다. 여기에는 손쉽게 구할 수 있는 일본의 간지 글자체를 `hlatex`에 덧붙여 사용할 수 있도록 하였고 문서 모양새도 좀더 한국적인 것을 사용할 수 있도록 하였으며 글자체가 늘어남으로써 불가피해 진 NFSS의 도입이 이루어졌다.⁶ 이 두 프로그램은 좀더 나은, 하나의 프로그램으로 일반에게 보급될 수 있고자 한글 라텍으로 통합되었는데 이로써 한글 라텍은 다음과 같이 변화하였다 (한글 라텍 버전 0.92e).

- ① 라텍 버전 2 ϵ 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 `hlatex`은 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 제공되었다. 따라서 문서의 인쇄는 300dpi와 600dpi의 해상도를 갖는 인쇄기에서만 가능했다. 그래서 한글 라텍 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 한글 라텍을 사용할 수 있게 되었다. 글자의 배치도 이전의 KAIST 식인 조합 완성 혼합식을 따르지 않고 일반적인 한국 표준 완성형 방식을 지향하게 되었다.

⁴유명한 한글 라텍 패키지들의 명칭은 다음과 같이 쓴다. `hTeXp`, `HTeX`, `Hangul-ucs`, `koTeX`.... 이곳에서 `hlatex`이라고 쓴 것은 `HTeX`의 전신으로 `HTeX`(한글 라텍)을 말하는 것이 아니므로 모두 소문자로 써서 구별하였다.

⁵그 당시 `hlatex`의 사용은 KAIST 이내와 해외에서만 허용되었다.

⁶NFSS는 라텍 버전 2 ϵ 의 주된 기능으로 New Font Selection Scheme의 약자이다. 라텍의 글자체 선택 방식을 대폭 개선함으로써 그 동안 라텍의 취약점으로 남아 있던 글자체 선택 문제를 완벽히 해결한 기능이다.

제 4 절 Omega(Ω)에 대해

현재 텍의 발전 방향을 바라보면 라틴 알파벳처럼 비교적 간단한 방법으로 처리하기 어려운 한글과 같은 문자의 처리에도 상당한 노력이 기울여지고 있음을 알 수 있다. 그러나 근본적인 문제는, 텍 자체가 7비트 부호 체계를 바탕으로 만들어져 있는 반면 한글 부호화 방식은 8비트의 문자가 두 번 연속되는 이중 바이트의 체계(KS X 1001, UCS2)이거나 세 번 연속되는 다중 바이트 체계(Unicode/UTF-8)라는 데에서 발생한다. 텍 버전 3⁷의 출현으로 유럽 언어와 같은 8비트 부호 체계의 처리는 해결되었지만 우리말과 같은 다중 바이트 부호를 처리하는 데에는 아직 많은 문제점을 안고 있다.

세계적인 추세도 각 나라의 모든 언어를 동시에 부호화 할 수 있는 통일된 부호화 방식을 지향하게 되었고, 국제 표준화 단체(ISO)는 32비트 코드 체계인 국제 부호화 문자 세트(UCS4)를 발표하였다. 이에 따라 텍에서 이 문자 세트를 사용할 수 있도록 하기 위해 8비트 코드 체계를 확장하여 16비트 코드 체계에 바탕을 둔 Omega(Ω)로 텍을 대체하는 시도가 나오게 되었다. Omega(Ω)는 16비트 텍의 구현으로서, 유니코드를 처리할 수 있는 텍이라고 간단히 표현할 수 있다. Omega(Ω)는 Yannis Haralambous와 John Plaice가 공동으로 개발하였는데 1996년 11월에 $\text{te}\text{\TeX}$ 버전 0.4를 바탕으로 시험판이 나온 이래, 1998년 3월에 발표된 web2c 버전 7.2에는 Omega(Ω) 버전 1.5를 공식적으로 채택하여 함께 보급하게 되는 단계에까지 오게 되었다. 그 후 Omega(Ω)와 ϵ - \TeX 을 결합한 Aleph(\aleph)가 \TeX Live에 포함되어 배포되고 있다. 비록 현재는 개발이 잠시 주춤하고 있는 듯하나, Omega(Ω)를 통한 실험은 유니코드를 이용하는 조판이라는 방향을 옳게 잡았던 것이라고 생각한다.

제 5 절 유니코드를 이용한 실험들 (2002-2006)

Omega(Ω)에 주목하기 시작하면서 한글 라텍은 EUC-KR 부호계밖에 표현할 수 없는 한계를 넘어서기 위해 Omega(Ω)의 \TeX 판인 Lambda 매크로를 개발하였다. 특히 2002년 KTUG을 중심으로 이루어진 일련의 실험과 그 성과인 u8hangul 스타일은 2005년의 한글 라텍 1.0.1의 형성에 결정적인 토대가 되었다. 한글 라텍 1.0.1은 여전히 EUC-KR 한글을 처리하는 \TeX 매크로와 더불어, 모든 한글(UHC)을 표현하고 식자할 수 있는 Lambda 패키지로도 개발되었던 것이다. “Omega(Ω)는 이제 거의 완숙한 단계에 도달하고 있고 앞으로는 텍을 완전히 대체하고 본격적으로 사용되는 방향으로 전개될 것을 희망하고 있습니다.”(은광희, 2005)

⁷현재의 \TeX 버전은 3.141592이다.

이 시기에 Lambda를 이용하여 한글 및 한자로 이루어진 고문헌의 식자에 성공한 일련의 실험은 KTUG 게시판에 그 흔적이 아직도 남아 있다. 이 때야말로 한글이라는 문자를 텍에서 표현하는 데 있어 한번의 도약을 이룬 시기였다고 볼 수 있다. 한글 표현의 한계가 없어진 것이다.⁸

그러나 Lambda를 일상적인 문서 작성에 쓰기에는 몇 가지 문제가 존재했다. 우선 pdf 최종 출력물을 얻는 중요한 방법인 pdfTeX을 사용할 수 없다는 것, 그 대신 DVIPDFMx⁹가 pdf 변환기 역할을 해줄 수 있었지만 Omega(Ω) 자체의 한계로 인하여 불편한 점들이 보고된 것, 그리고 2003년에 Omega(Ω)가 더이상 개발되지 않을 것이라는 소식을 듣게 된 것, \LaTeX 의 수많은 패키지 솔루션들을 그대로 사용하는 데 한계를 보인 것, Omega(Ω)를 위한 편리한 글꼴 세트가 충분히 마련되어 있지 않았던 것 등이었다.¹⁰

한편, pdf 문서 제작이 초미의 관심사로 떠오르면서, 한글 라텍스에 대한 약간의 추가적인 보완이 시도되기도 하였다. hyperref라는 pdf 제작에 필수적인 패키지와 hangul 패키지의 불일치를 해결하기 위하여 hangul-k라는 third-party 패키지가 제작되기도 하였으며, 한글 라텍스에 자간과 행간을 좀더 쉽게 적용하려는 일련의 노력들이 이루어지기도 하였다.

2004년과 2005년을 전후하여, 김도현과 김강수에 의해 hangul-ucs 패키지가 탄생한다. 이것은 Dominique Unruh의 latex-ucs 패키지를 기반으로 UTF-8 인코딩된 유니코드 한글 텍스트를 식자하는 것이었다.¹¹ \LaTeX 에서 유니코드 문자를 식자하는 방법으로 utf8.def와 utf8x.def가 있었는데, 이 가운데 latex-ucs 패키지는 utf8x.def를 포함하고 있는 것이었다. 이 패키지에는 한글 라텍스를 연구하고 수정한 결과가 반영되어 있었으며, 한글 문서화 서식은 거의 한글 라텍스의 것을 그대로 가져와서 사용하였다. 자동조사 처리나 행나눔과 같이 독자적인 연구의 결과도 물론 포함하고 있었다.

2005년과 2006년에 걸쳐, 더 완성된 한글 라텍스와 hangul-ucs 패키지는 각각 그 용도에 알맞게 발전하였다. 한글 라텍스의 지배적인 위치에는 변함이 없었으나, 빠르게 발전하고 있는 hangul-ucs도 점차 그 사용층을 넓혀가고 있었다.

⁸한글 윈도우즈 운영체제의 ‘확장완성형’(CP949)은 Lambda를 테스트하던 시기에 잠시 고려한 적이 없지 않으나, 원칙적으로 그 후 고려 대상에서 제외되었다. 유니코드를 이용하여 한글을 표현하는 것이 표준에도 부합하며 혼선을 최소화하는 방법이라는 데 동의했기 때문이다.

⁹KTUG의 조진환 교수가 개발한 dvi 드라이버 프로그램으로서, 이를 바탕으로 \XeTeX 의 xdvipdfmx가 만들어지기도 하였다.

¹⁰이런저런 이유로 인해, $koTeX$ 은 Omega(Ω)를 위한 지원을 제거하기로 결정하였다.

¹¹hangul-ucs 4.0 이후 버전과 현재의 $koTeX$ 은 latex-ucs에 의존하지 않는다.

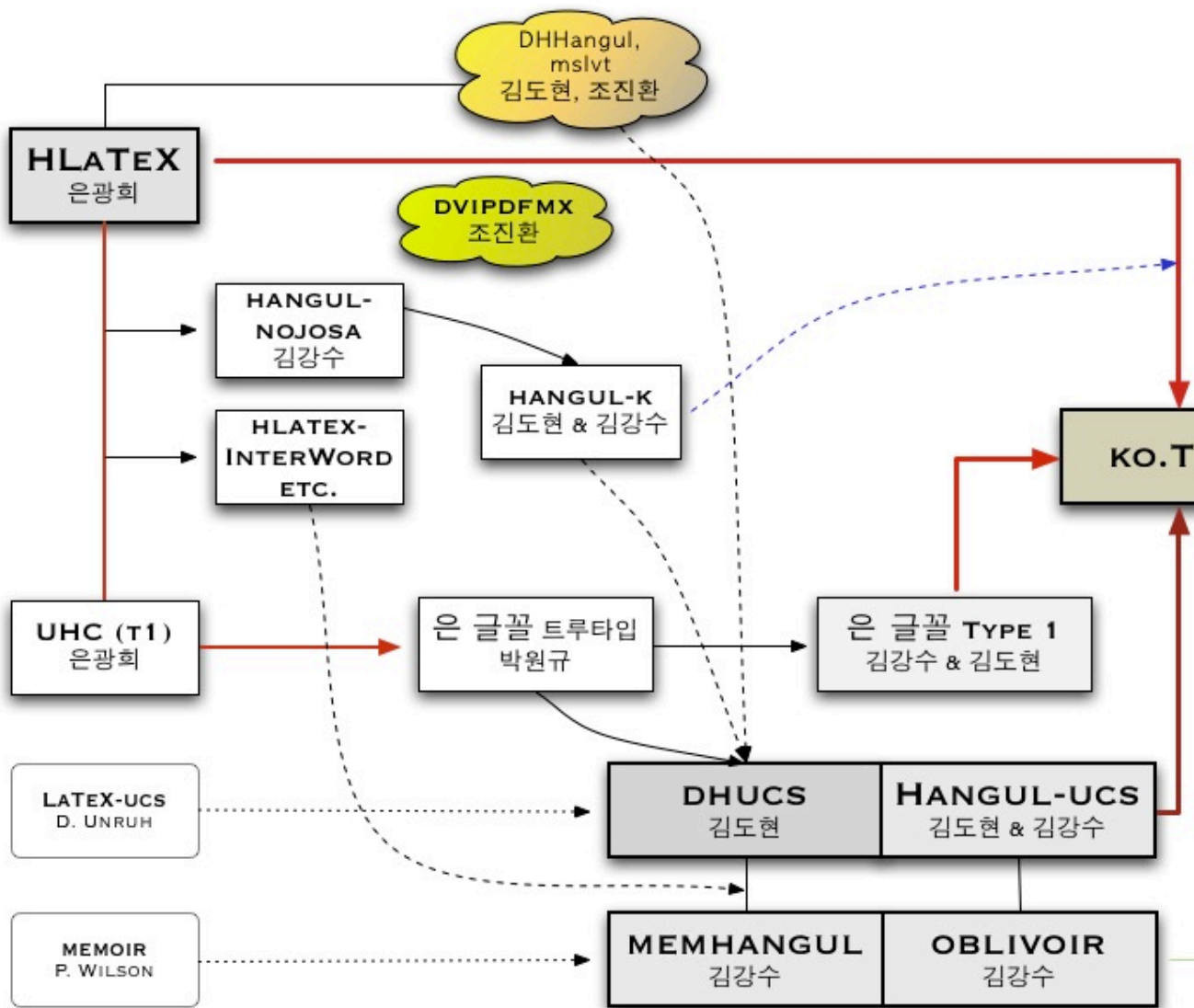


그림 1.1: ko.TeX의 탄생

제 6 절 ko.TeX의 탄생 (2007)

그림 1.1은 HLaTeX에서 출발하여 ko.TeX에 이르는 한글 텍의 발전 과정을 요약한 것이다.

2007년 6월 30일, 한국텍학회 모임에서, 은광희·김도현·김강수 등 한글 라텍·hangul-ucs의 저자들과 KTS 임원·회원 등이 모인 자리에서 이 두 한글 매크로 패키지를 통합하여 통합 한글 텍 패키지를 제작하기로 의견을 모은 것이다. 이것은 그 동안 한글 사용에 있어서 둘 이상의 패키지가 존재함으로 인해 빚어진 혼란을 불식하고 향후 한글 텍/라텍의 발전의 근거를 마련했다는 점에서도 중요한 일보전진이었고, 한글 라텍으로부터 오랜 시간 발전해 온 라텍에서의 한글 사용을 총체적으로 점검할 기회를 얻게 되었다는 점에서도 의미가 있는 결정이었다.

그로부터 일 개월 정도의 작업을 통하여 ko.TeX 베타 버전이 공개되었다. ko.TeX은 Unicode/UTF-8 인코딩과 EUC-KR을 모두 지원하기로 하였으나 원칙적으로 향후의 발전 방향은 Unicode로 합의하였다. 또한 은광희의 UHC 글꼴을 발전시킨 ‘은

글꼴 type 1'을 기본 글꼴로 채택하고 이를 EUC-KR 버전에도 적용시킴으로써, 폰트를 둘러싼 이런저런 불편한 점을 일소하였으며, pdf 제작 및 매크로의 호환성과 안정성에 있어서 몇 가지 중요한 진보를 이루었다.

제 2 장 설 치

제 1 절 \TeX 배포판

각 플랫폼별로 유명한 \TeX 배포판¹들이 있다. 리눅스에서는 Thomas Esser의 $\text{te}\text{\TeX}$ 이 부동의 지위를 오랜 동안 지켜왔으며, 매킨토시에서는 Classic Mac 시기의 $\text{Oz}\text{\TeX}$ 이 퇴조하고 Mac OS X 하에서 Gerben Wierda의 $\text{gw}\text{\TeX}$ 이 널리 쓰였다. 윈도우즈는 사정이 복잡하지만 $\text{MiK}\text{\TeX}$ 이 사실상 가장 많은 사용자를 가진 시스템이 되었으나 Fabrice Popineau의 $\text{fp}\text{\TeX}$ 도 한동안 우수한 사용자를 보유하고 있었다. $\text{MiK}\text{\TeX}$ 을 제외하면 이 모두가 web2c에서 파생된 시스템이라는 공통점을 지니고 있다. $\text{MiK}\text{\TeX}$ 도 종래에는 web2c를 변형해서나마 받아들이게 되었다.

그런데 2007년에 접어들면서 사정이 일변한다. 우선 $\text{te}\text{\TeX}$ 과 $\text{gw}\text{\TeX}$ 의 개발이 중단되었다. 또한 web2c 역시 독자적인 배포판(소스)로서 유지되지 않게 되었다. 그 대안으로 \TeX Live가 새로운 표준으로 등장했기 때문이다.

윈도우즈의 경우, 사정은 더욱 복잡해졌다. KTUG에서는 2006년, \TeX Live와 무관하게 독자적인 배포판 KTUG Collection을 제작하였는데, 이것은 web2c로부터 파생된 Akira Kakuto의 $\text{W32}\text{\TeX}$ 의 바이너리를 가져다 쓴 것이었다. 다행히도 \TeX Live 자체가 $\text{W32}\text{\TeX}$ 을 채택함으로써, KC2006 과 \TeX Live는 그 차이가 미미한 대단히 유사한 시스템이 되었다.²

이 글은 \TeX Live 2007을 기준으로 작성한다. 매킨토시와 리눅스 등 유닉스-류 운영체제에서는 \TeX Live 이외의 대안이 현재로서는 없다. 일부 리눅스 시스템에서 $\text{te}\text{\TeX}$ 이 여전히 주류를 이루고 있으나 조만간 \TeX Live로 이행하게 되리라고 기대한다. \TeX Live는 방대하기도 하거니와 대단히 안정적으로 동작하는 시스템으로, 사실상 다양한 배포판의 시대를 종결시킨 ‘유일’ 배포판의 지위를 차지할 것으로 예측된다.³

일부 상업용 \TeX 배포판의 경우는, 그 배포판을 판매하는 측에서 한글 사용에

¹ \TeX implementations

²내부적으로 차이점은 많다. 특히 한글 운영에 필요한 다양한 유틸리티를 제공하는 것, 일괄 인스톨러를 제공하는 것, 패키지 관리 유틸리티와 업데이트 유틸리티를 제공하는 것, 셸 유틸리티(KCmenu)를 제공하는 것 등은 \TeX Live보다 KC2006 을 더욱 편리한 시스템이 되게 한 면이 있다.

³ $\text{MiK}\text{\TeX}$ 에 대하여 별도로 언급하지 못하는 것은 유감이다. 그러나 $\text{MiK}\text{\TeX}$ 사용자들은 $\text{MiK}\text{\TeX}$ 에서 $\text{ka}\text{\TeX}$ 을 설치하는 적절한 방법을 찾아낼 것이라고 믿는다.

관한 해결책을 제시하는 것이 옳다고 생각한다. 이 글은 상업용 배포판에 대해서는 별도의 언급을 할 수 없을 것이다.

제 2 절 ϵ -TeX

koTeX/utf는 ϵ -TeX을 요구한다. ϵ -TeX의 확장 프리미티브를 활용하고 있기 때문이다. kotexplain 패키지도 tex으로는 컴파일되지 않으며 etex이나 pdftex을 이용하여야 한다.

다행히 TeXLive 2007에서 latex은 ϵ -TeX 엔진을 사용하도록 구성되어 있는 것이 기본값이므로 ϵ -TeX 사용에 있어서 사용자가 별달리 주의할 것은 없다.

다른 TeX 배포판, 예컨대 teTeX 등에서는 elatex이나 pdfelatex을 실행하지 않으면 안되는 경우도 있을 수 있다. 텍 시스템의 기본 ϵ -TeX 엔진을 ϵ -TeX으로 바꾸는 것은 쉽게 할 수 있으나 이 글의 범위를 넘어선다.

제 3 절 디렉터리 구조 (TDS)

표준적인 web2c 시스템 (TeXLive는 web2c 시스템이다)에서는 방대한 파일을 간단하게 찾기 위해서 kpathsearch라는 라이브러리를 이용한다. kpathsearch 라이브러리는 필요한 파일들이 일정한 디렉터리 구조에 따라 배치되어 있다고 가정하기 때문에 TeX 시스템에서 파일은 이러한 디렉터리 구조 (TDS⁴)를 따라 배치되어 있어야 한다. 이 글에서는 2004년 발표된 TDS version 1.1을 기반으로 설명한다.⁵

이 규정에 따르면, 텍에 필요한 파일들은 TDS의 출발점인 TEXMF 밑에 두게 된다. TEXMF는 텍의 원천 코드를 컴파일할 때 결정되며, 둘 이상을 둘 수 있다. TeXLive는 texmf, texmf-dist, texmf-doc라는 기본 TEXMF 트리와 더불어 사용자가 설정할 수 있는 texmf-local, \$HOME/texmf 등을 사용한다. 이들을 편의상 TEXMF root라고 부르기로 하겠다.

한 시스템에 여러 사용자가 있을 때, 시스템 전체에 적용되는 TEXMF들과 개별 사용자 자신이 이용하는 TEXMF도 분리할 수 있다. 이미 설치된 텍 운영 체제에서도 texmf.cnf 파일에서 TEXMF를 다시 설정할 수 있다. TDS를 준수할 경우, 서로 다른 텍 배포판에서도 TEXMF root의 경로만 다를 뿐이고 그 이하의 디렉터리 구조는 같다.

⁴TeX Directory Structure. <http://www.tug.org/tds/tds.pdf>

⁵MiKTeX 2.6은 TDS 1.1을 잘 따르지 않는 것으로 알려져 있다. 그러므로 아래의 설명은 MiKTeX에 적용되지 않는다.

ko \TeX 은 위의 어떤 TEXMF root 아래 설치되어도 무방하다. 이 위치를 편의상 [texmf] 라고 표시하겠다. 예컨대 /usr/local/texlive/texmf-local 을 ko \TeX 이 설치될 위치로 선정할 수 있을 것이다. 되도록이면 .../texlive/2007/texmf 이나, texmf-dist와 같은 시스템 디렉터리와는 분리하는 것이 좋다.

‘Top-level directories’는 텍 운영 체계의 구성 요소로서 tex (텍 파일), fonts (폰트 관련 파일), metafont (폰트가 아닌 METAFONT 파일), metapost (METAPOST 파일), bibtex(Bib \TeX 파일), scripts (운영 체계 독립적인 실행 파일), doc (사용자용 문서 파일), source (원천 파일), <구현> (emtex, vtex, web2c와 같은 텍 구현 관련 파일), <프로그램> (mft, dvips와 같은 프로그램에 따른 입력 및 설정 파일) 등등이 해당된다.

텍 체계를 구성하는 모든 명령(매크로) 파일의 위치는 다음과 같은 계층 구조를 갖는다.

TEXMF/tex/틀/꾸러미

‘틀’은 텍의 용어로 \dump 되는 것들을 말하는데, ko \TeX 은 플레인텍(plain \TeX)을 위한 kotexplain, 라텍을 위한 ko \TeX -euc/utf 등으로 이루어지므로, 거기에 맞게 [texmf]/tex/plain/kotex 또는 [texmf]/tex/latex/kotex 등의 위치에 놓이게 된다.

글자체는 다음과 같은 규정에 의해 설치할 것을 요구한다.

TEXMF/fonts/형식/제공자/글자체

TEXMF/fonts/{enc,lig,map}/하위경로

‘형식’은 글자체의 종류를 가리킨다. 즉, tfm, vf, type1, afm 등이 이에 해당한다. 글자체의 부호화나 합자 혹은 글자체 이름 대응 파일들은 각각 enc, lig, map ‘형식’ 디렉터리에 평행한 디렉터리에 따로 둔다. ‘제공자’의 대상은 글자체를 만드는 업체나 만든이 등인데, adobe 회사의 type 1 글자체 (adobe), bitstream 회사의 type 1 글자체 (bitstrea), D. Knuth 교수의 METAFONT 글자체 (public), J. Knappen 씨의 METAFONT 글자체 (jknappen) 등을 예로 들 수 있다. ‘글 자체’는 글자의 모양에 따른 이름이다. adobe 회사에서 만든 글자체에는 times, helvetic 등의 이름이 사용되고 D. Knuth 교수가 텍 운영 체계를 위해서 만든 글자체의 경우는 cm, ec 등의 이름이 사용되며 ko \TeX 에서 제공하는 글자체에서는 myoungjo, gothic 등의 이름이 사용된다.

ko \TeX 의 글자체들은 다음처럼 위치하게 된다.

[texmf]/fonts/형식/kotex/{base,extra}/{myoungjo,gothic,...}
[texmf]/fonts/map/dvips/kotex

제 4 절 일반적인 설치 지침

4.1 다운로드

ftp://ftp.ktug.or.kr/KTUG/ko.TeX/ 디렉터리에서 배포되는 압축 파일은 다음 두 개이다.

- ko.tex-macro.tar.gz
- ko.tex-unfonts-base.tar.gz

둘 다 .tar.gz로 압축된 파일로서 gunzip과 tar를 이용하여 압축해제할 수 있다.

4.2 매크로와 유틸리티

먼저 *-macro-* 압축파일을 적당한 곳에 풀면, 몇 개의 파일 (INSTALL, LICENSE 등)과 일곱 개의 하위 디렉터리를 얻을 수 있다. 이것을 다음과 같이 이동한다.

=====	
subdirectories and/or files	destination
=====	
bibtex/euc	[texmf]/bibtex/bst/kotex/euc
doc/*	[texmf]/doc/latex/kotex/*
latex/*	[texmf]/tex/latex/kotex/*
makeindex/*	[texmf]/makeindex/kotex/*
plain/*	[texmf]/tex/plain/kotex/*
sfd/*	[texmf]/fonts/sfd/*
util/*.pl	[texmf]/scripts/kotex/perl/*.pl
=====	

util 디렉터리 안에는 두 개의 .c 소스와 몇 개의 *.pl 스크립트가 있다. 이 중 .c는 컴파일하고⁶ *.pl은 실행 권한을 부여하여 /usr/local/bin/으로 이동한다.⁷

hbibtex.c

```
$ gcc -o hbibtex hbibtex.c
$ cp ./hbibtex /usr/local/bin/
```

⁶ 윈도우즈용으로 .exe 파일이 함께 제공된다.

⁷ 윈도우즈에서는 Active Perl 등을 설치하여 이 스크립트를 실행 가능하게 해두어야 한다. 혹은 설치 패키지를 이용하는 경우 .exe 형식 파일로 제공하는 경우도 있을 것이다.

```
hmakeindex.c
$ gcc -o hmakeindex hmakeindex.c
$ cp ./hmakeindex /usr/local/bin/
*.pl
$ chmod a+x *.pl
$ cp ./*.pl /usr/local/bin/
```

4.3 글자체의 설치

`*-fonts-base-*` 압축 파일을 내려받은 다음 적당한 곳에 풀면 라이선스 관련 파일 몇 개와 함께 두 개의 하위 디렉터리가 있다. 이것을 아래와 같이 이동한다. 기본으로 배포되는 폰트 패키지에는 명조, 고딕, 타자, 그래픽, 네 글꼴 가족만 포함되어 있다. 이밖의 ‘은 글꼴’ 시리즈는 `extra` 폰트라는 이름으로 배포되는데, 만약 많은 글자체를 필요로 한다면 `extra` 패키지까지 설치해야 할 것이다. 설치 방법은 동일하다.

```
=====
directory                destination
=====
fonts                    [texmf]/fonts
tex                      [texmf]/tex
=====
```

이 글자체들에 대한 정보를 텍 엔진에게 알려주는 것이 `updmap`을 통한 `map`의 생성이다. `TEXLive 2007`은 이 과정을 매우 간편하게 진행할 수 있는데, 다음 절차를 따르면 된다.

```
$ mktexlsr
$ updmap-sys --enable Map=kotex-base.map
```

만약 `super user` 권한을 가지고 있지 않다면 `updmap-sys` 대신 `updmap`을 실행한다.
`extra` 폰트 패키지도 설치했다면

```
# mktexlsr
# updmap-sys --enable Map=kotex-extra.map
```

이것으로 끝이다.

제 5 절 각 플랫폼별 설치 패키지

앞 절에서 설명한 일반적 설치 지침을 따르는 것이 번거롭고 불편할 수도 있으므로 그럴 때는 간편한 각 배포판, 플랫폼별 설치 패키지를 이용하는 방법이 있다. `install package`⁸ 디렉터리에 가면 윈도우즈 KC2006 및 매킨토시 TeXLive를 위한 설치 패키지가 마련되어 있다. 리눅스는 각 배포판별로 간편한 설치 유틸리티를 제작할 수 있을 것인데, 이에 대한 정보는 KTUGFaq::ko.TeX/instpack 페이지⁹에서 찾을 수 있다.

제 6 절 CVS 개발 버전

CVS::ko.TeX¹⁰에서 최신의 개발 버전을 확인할 수 있다. 4.1 절에서 언급한 파일들은 CVS snapshot을 정기적으로 릴리스한 것이므로 비교적 안정적이거나, CVS 개발 버전은 사용자가 개발에 참여할 의사가 있거나 최신 버전을 활용하고자 할 때, cvs를 통하여 다운로드받을 수 있을 것이다.

anonymous로 CVS를 리트리브하려면 다음과 같이 한다.

```
$ cvs -d :pserver:anonymous@cvs.ktug.or.kr:/home/cvsroot login
$ cvs -d :pserver:anonymous@cvs.ktug.or.kr:/home/cvsroot \
  checkout ko.TeX
```

패스워드를 물어오면 그냥 엔터를 치면 진행할 수 있다.

⁸<http://ftp.ktug.or.kr/KTUG/ko.TeX/instpack/>

⁹<http://faq.ktug.or.kr/faq/ko.TeX/instpack>

¹⁰<http://cvs.ktug.or.kr/viewcvs/ko.TeX>

제 3 장 사용

한글 라텍 ko_{TeX} 은 은광희의 $H_{\text{M}}\text{TeX}$ 과 김도현·김강수의 $hangul\text{-ucs}$ 를 합쳐서 하나의 패키지로 구성한 것이다. 1990년대 이래 한글 사용의 사실상 표준이었던 이 두 패키지가 결합함으로써 명실공히 라텍에서의 한글 사용에 새로운 장을 열었다고 평가해도 좋을리라.

ko_{TeX} 은 본격적인 한국어(한글) 텍 시스템이다.¹ 라텍은 물론이고 플레인텍에서 컨텍스트에 이르기까지 거의 모든 텍 관련 매크로에서 한글을 구현하도록 할 수 있다. 그래도 그 중심은 역시 한글 ‘라텍’일 것이다. 이 글은 한글 ‘라텍’에 대해서 주로 다루고 다른 매크로에서의 한글 구현에 관한 사항은 별도의 장으로 설명하겠다. 앞으로 별다른 언급이 없으면 ko_{TeX} 이 한글 라텍 매크로를 (주로) 가리키는 것으로 하자.

한글 라텍으로서 ko_{TeX} 은 EUC-KR 코드와 UTF-8 한글 코드를 지원한다.² 그 가운데 EUC-KR 지원은 일종의 “하위호환성” 측면에서 제공되는 것으로, 향후 발전은 UTF-8 부분만 반영될 것이다. EUC-KR 코드로 작성된 기존의 한글 문서를 컴파일할 수 있고 이것을 더 쉽게 UTF-8 문서로 바꿀 수 있게 하는 것이 두 코드를 모두 지원하는 원래의 목적이며 EUC-KR 버전은 UTF-8 버전과 항상 동일하게 동작하지 않는다. 또한 EUC-KR 한글 문서는 비교적 쉽게 UTF-8로 포팅할 수 있으나 그 역의 경우는 잘 되지 않거나 많은 노력이 필요할 수 있다.

그럼에도 불구하고 ko_{TeX}/euc 는 $H_{\text{M}}\text{TeX}$ 에 비해서 다음과 같은 점이 개선되었다. 주로 pdf 제작과 관련된 것들이 많다.

euc

1. pdf 책갈피를 만들어도 한글이 “깨지지” 않게 할 수 있다.
2. pdf 문서에서 한글 텍스트의 검색과 추출이 되게 할 수 있다.
3. KTUG에서 발전한 한글 타이포그래피의 성과를 부분적으로나마 이용하여 비교적 잘 정돈된 한글 문서를 만들어낸다. 그러나 utf 버전은 더욱 발전된 타이포그래피를 다루고 있으므로 되도록이면 utf 버전이 사용되기를 희망한다.

¹ ko_{TeX} 은 텍스트 상황에서 $ko_{\text{.TeX}}$ 이라고 쓰고 “케이오텍” 또는 “코리언 텍”이라고 읽는다. ko 는 한국어를 의미하는 국제부호에서 취한 것이다. ko 를 소문자로 쓴다.

²윈도우즈에 친숙한 CP949 코드는 지원하지 않는다. CP949는 사실상 유니코드의 현대 한글 음절 영역에서 다룰 수 있는 모든 문자에 해당하므로 유니코드를 통하여 지원하는 것이 바람직하다고 판단하였다.

한편, hangul-ucs 에 비해서는 주로 (linux 에서의) 설치와 관련된 문제가 현저히 개선되었다.

utf

1. latex-ucs 패키지에의 의존성이 없다.
2. ttf2pk 를 사용하지 않음으로써 ttf2pk 에 대한 의존성도 사라졌다.
3. H_WTeX 과 통합되었으므로 hlatex 에 대한 의존성도 필요없다.
4. 트루타입 ‘은 글꼴’을 사용하지 않기 때문에 별도로 폰트를 설치해야 하는 불편이 없다.
5. dvipdfmx.cfg 나 ttf2pk.cfg 등의 설정 파일 수정을 일일이 해야 하는 불편이 없어지고 updmap 에 의한 일괄 명령 한 번으로 모든 폰트 설정이 이루어진다.
6. 기본 패키지의 크기가 이전에 비하여 현저히 줄었다.

제 1 절 기본 사용법

ko_WTeX 의 기본 스타일은 kotex.sty 이다. 다음과 같이 쓴다.

`\usepackage{kotex}`

utf|euc

언어 코드를 첫번째 인자로 지정하는데, 디폴트는 utf 이다. 그러므로 utf 는 별도로 명시하지 않아도 되지만 euc 문서를 작성할 때는 반드시 euc 옵션을 되도록 맨처음에 지정한다.

`\usepackage[euc]{kotex}`

명령	대응
<code>\usepackage{kotex}</code>	<code>\usepackage{dhucs}</code>
<code>\usepackage[hangul]{kotex}</code>	<code>\usepackage[hangul]{dhucs}</code>
<code>\usepackage[euc]{kotex}</code>	<code>\usepackage{hfont}</code>
<code>\usepackage[euc,hangul]{kotex}</code>	<code>\usepackage{hangul}</code>

H_WTeX 에는 두 가지 사용법이 있었다. 하나는 hfont 이고 다른 하나는 hangul 이었다. ko_WTeX 에서는 한글 서식 hangul 옵션이 주어진 상태와 그렇지 않은 상태로 구분한다. 영문 위주 문서에 한글을 사용할 때는 hangul 옵션 없이 쓰고, 한글 위주의 문서에서 한글식 서식을 설정하려면 hangul 옵션을 지정한다.

kotex.cfg 사용자는 일반적으로 utf나 euc 가운데 한 가지를 주로 쓰지 둘 다 쓰게 되는 경우는 거의 없을 것이다. kotex.cfg 파일은 이럴 경우에 대비하여 자신이 쓰는 kotex 옵션을 미리 지정해두고 단순히 `\usepackage{kotex}` 이라고만 하여 그 옵션들을 활성화하게 할 수 있다. 이 파일은 `[texmf-ktug]/tex/latex/kotex/`에 있으므로 자신에게 맞게 적절하게 수정해서 쓰도록 한다.

제 2 절 pdf 문서 작성을 위한 설정

2.1 ko.TeX/utf 문서

utf

utf 문서는 다음과 같이 한다. 유니코드 문서에서는 hyperref에 `[unicode]` 옵션을 지정하는 것으로 pdf bookmark 문제가 모두 해결되므로, .out 파일의 코드 변환과 같은 문제에 신경쓸 필요가 없다.

```
\usepackage{kotex}
\usepackage{ifpdf}
\newcommand\myDRIVER{\ifpdf pdftex\else dvipdfm\fi}
\usepackage{\myDRIVER,unicode,bookmarks}{hyperref}
```

여기서는 `[unicode]` 옵션이 반드시 지정되어야 한다. 그리고 CJKbookmarks를 쓰지 않는 점이 다르다. 또한 euc-bookmark와 같은 번거로운 절차가 필요없다. 위의 예에 해당하는 컴파일 절차 예시는 다음과 같다.³

```
$ (pdf)latex foo
$ bibtex foo
$ komkindex -s kotex foo
$ (pdf)latex foo
$ (dvipdfmx foo)
```

pdf 문서의 텍스트 검색과 추출은 매우 중요한 문제이다. 이것은 어떤 드라이버로 pdf를 생성하느냐에 따라 조금 달라지는데, 일괄 요약하면 다음과 같다.

dvipdfmx 기본 폰트인 type 1 글꼴을 매우 잘 처리하고 이로부터 검색 및 추출이 가능한 pdf를 만들어낸다. 별도의 조치가 필요없다.

dvips 트루타입을 처리하지 못하는 점 때문에 이전부터 문제가 있다고 생각되었으나 적어도 ko.TeX이 제공하는 폰트를 사용하는 한 검색 추출에 문제는 없다. 역시 별도의 조치가 필요없다.

³komkindex는 Perl Script 이므로 경우에 따라 komkindex.pl을 실행해야 할 것이다. 이후로도 같다.

pdfTeX 이 때는 `\pdfgentounicode`를 활성화해주어야 한다.

그러므로, `preamble`에 다음과 같이 선언하는 것으로 될 것이다.

```
\ifpdf
  \input glyphtounicode
  \pdfgentounicode=1
\fi
```

어떤 드라이버를 사용하든 모두 포스트스크립트 윤곽선 글꼴이 사용되고, pdf 북마크가 잘 만들어져 있으며 한글 텍스트의 검색 및 추출이 자유롭다.

2.2 ko.TeX/euc 문서

euc

```
\usepackage[euc]{kotex}
\usepackage{ifpdf}
\newcommand\myDRIVER{\ifpdf pdftex\else dvipdfm\fi}
\usepackage[\myDRIVER,CJKbookmarks]{hyperref}
```

위의 것은 euc 문서를 위한 설정이다. `hyperref`는 거의 필수이므로 이를 로드 하여야 하고, 북마크는 `CJKbookmarks` 옵션을 준다. 이제 이 문서를 처리하는 과정에서, 맨 마지막 **TeX** 컴파일 직전에 `euc-bookmark` 유틸리티를 한번 실행해주면 된다. 이 유틸리티가 하는 일은 북마크를 위한 파일의 인코딩을 `hyperref`가 이해하는 유니코드 인코딩으로 바꾸어주는 일이다. 그러므로 컴파일 과정을 예시하면 다음과 같이 될 것이다.⁴

```
$ pdflatex foo
$ hbibtex foo
$ komkindex -euc -s hind foo
$ euc-bookmark.pl foo.out >foo-a.out
$ cp foo-a.out foo.out
$ pdflatex foo
```

DVIPDFMx를 이용하려 하는 경우 불편한 `euc-bookmark`를 쓰지 않아도 한글 북마크를 잘 만들어준다. 이 때는 다음과 같은 코드가 필요하다.

```
\ifpdf\else
  \AtBeginDvi{\special{pdf: tounicode KSCms-UHC-UCS2}}
\fi
```

⁴KC2006을 위한 설치 패키지에서는 `komkindex.exe` 등을 제공하기도 한다. 이 때는 굳이 `.pl`과 같은 확장명을 붙이지 않아도 된다. 또한 `cp` 명령은 윈도우즈에서는 `copy`이다.

그리고 다음 순서로 컴파일하면 된다.

```
$ latex foo
$ hbibtex foo
$ komkindex -euc -s hind foo
$ latex foo
$ dvipdfmx foo
```

이것은 오직 DVIPDFMx 를 쓰는 경우에만 유효하다.

2.3 요약

이상을 요약하면, 다음과 같다.

UTF:

```
\documentclass{article}
\usepackage{kotex}
\usepackage{ifpdf}
\ifpdf
  \usepackage[unicode]{hyperref}
  \input glyphtounicode\pdfgentounicode=1
\else
  \usepackage[unicode,dvipdfm]{hyperref}
\fi
```

EUC:

```
\documentclass{article}
\usepackage[euc]{kotex}
\usepackage{ifpdf}
\ifpdf
  \usepackage[CJKbookmarks]{hyperref}
  \input glyphtounicode\pdfgentounicode=1
\else
  \usepackage[dvipdfm,CJKbookmarks]{hyperref}
  \AtBeginDvi{\special{pdf: tounicode KSCms-UHC-UCS2}}
\fi
```

이와 관련하여 다음 두 가지를 지적해 두기로 한다.

1. 위의 코드가 제대로 동작하려면 KSCms-UHC-UCS2 라는 이름의 cmap 이 있어야 한다. 이 cmap 은 ghostscript 의 resource 디렉터리에 들어 있으므로, 만약 cmap 을 찾는 데 실패하는 에러를 만난다면 자신의 T_EX 시스템에게 이 cmap 위치를 알려주도록 하라.
2. 위의 pdfgentounicode 에 관련된 것은 type 1 글꼴이 사용되었을 때 유효하다. 사용자가 자신의 개인적인 용도로 트루타입으로 조판하려 한다면 이 코드 보다는 \usepackage{dhucs-cmap} 을 선언하도록 하여야 한다.

제 3 절 폰트에 대하여

ko.T_EX의 가장 큰 특징이 폰트에 관련된 부분이다. 기본 글꼴로 “은 글꼴 type 1”을 채택하였는데 이것은 H_EL_AT_EX의 기본 글꼴이었던 UHC 글꼴을 트루타입으로 바꾼 은 글꼴 (hangul-ucs 의 기본 글꼴)로부터 얻어진 type 1 이다.

원래 UHC 글꼴이 type 1 이었는데 왜 은 글꼴로부터 별도로 type 1 글꼴을 만들어내어야 했던가? 그것은 UHC 글꼴이 가진 독특한 특성, 즉 자소 글꼴이라는 점 때문이었다. UHC 폰트 자체는 일종의 자소 글꼴로서, 이것을 실제의 문자에 대응시키기 위해서는 vf를 통하여 자소 조합을 실행해야 한다. 각 문자에 유니코드 코드 포인트를 대응시켜주어야 텍스트의 검색 추출이 가능해질 것이므로 UHC 폰트 자체만으로는 이것을 하는 것이 어려웠다. 반면 은 글꼴 type 1은 유니코드 배열에 따라 완성된 문자를 가지고 있는 글꼴이다. 은 글꼴 type 1 자체가 UHC에 기원을 두고 있으니만치, “은 글꼴 type 1은 UHC의 자소를 미리 조합하여 유니코드 코드 포인트를 부여한 글꼴”이라고 이해할 수 있으리라 생각한다.

UHC 글꼴 (H_EL_AT_EX) → 은 글꼴 (hangul-ucs) → 은 글꼴 type 1 (ko.T_EX)

3.1 기본 글꼴

기본 (base) 폰트는 명조(바탕), 돋움(고딕), 타자, 굴림의 네 종류이다.

글꼴	file	name(utf)	name(euc)	한자	볼드	변형 (c,o)
명조	outbtmXX	utbt	mj (wmj)	O	O	O
고딕	outgtmXX	utgt	gt (wgt)	O	O	O
타자	outtzmXX	uttz	tz (wtt)	X ⁵	X	X
그래픽	outgrmXX	utgr	gr (wgr)	X	O	O

⁵고딕의 한자를 가져다 쓴다.

euc와 utf 양쪽에서 모두 동일한 글꼴을 사용하는데, 다만 euc에서는 vf를 통하여 필요한 글자를 뽑아서 완성형 인코딩에 맞게 재배열하여 사용한다.

3.2 추가 글꼴

추가 폰트(extra)에는 은 글꼴의 나머지 폰트들이 들어 있다. 한자 등이 들어 있거나 변형 글꼴이 제공되는 것은 다음과 같다.

글꼴	file	name(utf)	name(euc)	한자	볼드	변형 (c,o)
신문	outshmXX	utsh	sh (wsh)	O	X	O
옛글	outytmXX	utyt	yt (wyt)	O	X	O
궁서	outgsmXX	utgs	gs (wgs)	O	X	O
필기	outpgmXX	utpg	pg (wpg)	X	O	O
봄	outbmmXX	utbm	bm (wbm)	X	X	O

다음 추가 글꼴은 굵은 글꼴, 기울인 글꼴, 장평줄인 글꼴 등의 변형 서체를 전혀 지원하지 않고 오직 기본 모양 하나만으로 제작된 것들이다.

글꼴	file	name(utf)	name(euc)	한자	볼드	변형 (c,o)
자모명조	outjbtmXX	utjbt	jbt	X	X	X
자모고딕	outjgtmXX	utjgt	jgt	X	X	X
자모노벨	outjnmXX	utjnv	jnv	X	X	X
자모소라	outjsrmXX	utjsr	jsr	X	X	X
펜글씨	outpnmXX	utpn	pn	X	X	X
펜흘림	outphmXX	utph	ph	X	X	X

이상 15종의 폰트가 트루타입 은 글꼴 시리즈에 대응한다. 여기에 H₂TeX 1.0에서 새로 도입된 세 종류 글꼴 중 두 종류를 추가 글꼴(extra-hl)로 제공한다. 이 추가글꼴들도 기울임/장평줄임 등의 변형 글꼴 없이(즉, m-시리즈만으로) 제공된다.

글꼴	file	name(utf)	name(euc)	한자	볼드	변형 (c,o)
디나루	outdnmXX	utdn	dn (wdn)	X	O	X
필기a	outpgamXX	utpga	pga	X	X	X

참고로, euc에서 extra-hl 폰트를 사용하려면 \usepackage{extra-hlfont}를 지시해주어야 한다.

euc

각 폰트의 인쇄된 모양을 보려면 부록을 참고하라.

3.3 은 글꼴 트루타입 폰트 세트

hangul-ucs에서 사용하던 은 글꼴 트루타입 폰트도 사용이 가능하다. 이것은 별도로 설치를 하여야 하며, 은 글꼴 트루타입도 별도로 시스템에 등록하여야 한다.

utf

utf에서는 일반적인 방법으로 이 폰트를 사용할 수 있다. 다만 폰트 명칭이 ut..가 아니라 un..이라는 점을 알아두자. 따라서 예컨대

```
\SetHangulFonts{unbt}{ungt}{untz}
```

```
\SetHanjaFonts{unbt}{ungt}{ungtz}
```

와 같이 선언하여 은 글꼴 트루타입을 쓰도록 할 수 있다.

euc

euc에서 은 글꼴 트루타입을 쓰려면 `\usepackage{unttf}`를 선언한다.

트루타입 사용은 DVIPDFMx로 pdf를 제작할 때 유용하다. pdfTeX에서 사용하지 못하는 바는 아니지만 기울인 글꼴 등이 잘 처리되지 않는다. 은 글꼴 type 1과 은 글꼴 트루타입은 사실상 같은 글꼴이라고 보아도 무방하다.

3.4 타자 글꼴에 관한 주석

은타자는 현재 거의 유일한 고정폭(monospaced) 글꼴이다. 이 글꼴에 대해서 그 디자인을 두고 논란이 없는 것은 아니지만 그래도 `\ttfamily`에 대응하는 고정폭 글꼴의 필요성을 고려한다면 매우 중요한 글꼴이라고 할 수 있다.

고정폭으로 쓰일 것이므로 볼드나 기울임 등은 불필요하다. 또한 폭을 변경시키는 변경폭(c-시리즈, x-시리즈)도 제공될 필요가 없다. 타자 글꼴 자체에 한자가 들어 있지 않은데, koTeX의 타자 글꼴은 `uttz`로 지정해도 가상 글꼴을 통해 고딕(돋움)의 한자 부분을 타자체의 폭에 맞게 조절하여 사용한다. 은 글꼴 트루타입은 그렇지 않으므로 위의 예와 같이 `ungtz`로 해야 한다.

3.5 요약

koTeX은 “은 글꼴 type 1”을 기본 폰트로 한다.

이로부터 얻을 수 있는 이점은 다음과 같다.

1. pdfTeX에 친화적인 폰트 세트이다. pdfTeX과 DVIPDFMx, 심지어 dvips까지 모두 이 형식의 폰트를 잘 다루므로 범용성과 호환성이 매우 높아졌다. pstricks를 위해서 폰트를 바꾸어야 하는 불편을 없앴다.
2. TeX 배포판의 유용한 유틸리티 updmap을 이용하여 설치하므로 설치 과정이 깔끔하게 끝난다.

3. pdf_T_EX, DVIPDFMx, dvips 어떤 것으로도 한글 텍스트의 검색과 추출이 되는 pdf를 제작할 수 있게 되었다.
4. 폰트의 품위는 은 글꼴 트루타입과 마찬가지로이며, pdf에 임베드되었을 때도 훌륭하다.

생각해볼 수 있는 결점은, pdf의 폰트 정보를 확인했을 때 수백 개의 outbtmac, outbtmad... 하는 식으로 확인되는 것이 좀 거슬릴 수 있고, 임베드된 pdf의 크기가 약간 커지는 정도인데, 둘 다 치명적인 문제는 아니라고 생각한다. 파일 크기는 무시할 수 있을 정도이며 DVIPDFMx를 쓰면 현저히 줄일 수 있고, 폰트 정보는... pdf를 제작하는 목적이 폰트 정보를 제공하기 위해서는 아니지 않겠는가.

3.6 폰트의 라이선스

은 글꼴 type 1 15종 (base, extra)은 GNU GPL로 배포된다. extra-hl 2종 폰트는 ko_T_EX의 라이선스(LPPL)를 따른다. 은 글꼴 type 1을 내장(임베드)한 pdf의 라이선스 문제에 대해서는 김도현의 글(김도현, 2007)을 참고하라. 결론은 GPL인 은 글꼴을 임베드하였다 해도, GPL의 감염성이 텍스트의 내용에 미치는 것은 아니라는 것이다.

제 4 절 utf 버전과 euc 버전의 관계

원칙적으로 hangul-ucs에서 계승한 ko_T_EX/utf 버전과 H_E_T_EX에서 이어받은 ko_T_EX/euc 버전은 상호 독립적이다. 또한 그 출력 결과가 완전히 동일하지 않으며 특히 폰트 선택과 관련된 부분은 명령 간의 호환성을 현단계에서는 보증하지 못한다. 그 이유는, ko_T_EX/euc의 의미를 “하위호환성” 측면에 비중을 두었기 때문이다. 이와 더불어 완성형 한글 코드를 처리하는 방식에 있어서 UTF-8을 처리하는 방식(_E_T_EX 표준)과 다른 점이 엄연히 존재하며 이를 호환가능하게 만드는 데 드는 노력에 비해 그 성과가 우리가 지향하는 바에 비추어 크지 않다고 판단한 점도 작용하였다.

1. 폰트 선택 명령 체계가 서로 다르다. ko_T_EX/euc는 H_E_T_EX의 HFSS를 그대로 이어받고 있다. 즉 예컨대 \hfontfamily나 \hfontseries, 심지어 \pgfamily와 같은 명령이 제공된다. 반면 ko_T_EX/utf는 이러한 종류의 명령은 제공하지 않으며 다만 한글 글꼴 가족을 선택할 수 있는 몇 개의 일괄 명령만이 제공된다.

2. 행자름 방법이 다르다. $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 는 (약간의 개선이 없지 않으나) 여전히 한글 앞에 영문자 등에 잇대어 있을 때 행이 나누어지지 않을 수 있다.
3. 미세간격과 같은 새로운 타이포그래피가 $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 에는 적용되지 않는다. 즉 `[finemath]` 옵션이 하는 일을 $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 는 전혀 하지 않는다.
4. 식자할 수 있는 문자의 범위가 다르다. 원칙적으로 $\text{ko}\text{T}_{\text{E}}\text{X}/\text{utf}$ 는 문자 범위의 제한이 없으나, $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 는 EUC-KR 완성형 한글로 그 범위가 제한된다.
5. $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 는 항상 `nonfrenchspacing`으로만 조판한다. `frenchspacing`과 유사한 효과를 얻으려면 마침표 뒤의 공백을 일반 공백과 같도록 맞추어야 할 것이다. 반면 $\text{ko}\text{T}_{\text{E}}\text{X}/\text{utf}$ 는 `nonfrenchspacing`을 옵션으로 선택할 수 있다.
6. $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 로는 다국어 문서를 조판할 수 없거나 더 많은 노력이 필요하다.
7. `oblivoir`나 `memhangul`은 `euc` 버전이 없다.

궁극적으로 한글 텍은 유니코드를 지향할 수밖에 없다. 그러므로 향후 한글 텍의 개선과 발전에 대한 노력도 $\text{ko}\text{T}_{\text{E}}\text{X}/\text{utf}$ 를 중심으로 이루어질 수밖에 없을 것이다. $\text{ko}\text{T}_{\text{E}}\text{X}/\text{euc}$ 는 예외적으로 부득이하게 필요한 경우에 한하여 사용되기를 희망한다.

제 II 편

ko.T_EX/utf

제 4 장 유니코드 문서 작성하기

제 1 절 개관

ko_{TeX}/utf은 UTF-8 유니코드로 입력되는 한글을 \TeX 문서에서 식자하고 조판하게 하려는 스타일 패키지로서, 그 핵심 부분인 dhucs 패키지를 김도현이 작성하였다.

제 2 절 첫번째 문서

2.1 간단한 문서 작성

그림 4.1 과 같은 간단한 문서를 작성해보자.

ko_{TeX}/utf의 기본 스타일은 kotex 패키지이다. 그러나 이 스타일은 dhucs 패키지를 부른다. 또한 대부분의 contrib 패키지의 명칭은 dhucs-로 시작한다.

제 10 행 이후에서 hyperref 패키지를 로드한다. 반드시 [unicode] 옵션을 주어야 한다. 한글 PDF 책갈피(bookmarks) 설정은 hyperref의 default 이므로 별도로 지정하지 않아도 책갈피가 만들어진다. 만약 책갈피를 만들지 않고자 한다면 bookmarks=false를 옵션에 추가한다. 패키지를 로드한 후 \hypersetup 명령을 이용해서 설정값을 추가할 수도 있다.¹

제 9 행 이후의 \ifpdf .. \else .. \fi 구문은 pdf_{TeX}의 실행 여부에 따라 서로 다른 설정을 하는 부분이다. 이 구문을 사용하려면 ifpdf 패키지가 필요하다. 제 12 행은 pdf_{TeX}에서 텍스트의 검색·추출을 가능하게 하는 행이다. DVIPDFMx를 위해서는 별도의 설정이 필요없다.

제 4 행부터 제 7 행까지는 반드시 필요한 경우가 아니면 꼭 지정하지 않아도 상관없다. 각 (부가) 패키지의 작용 등은 이 문서를 읽으면 알 수 있다. 제 5 행은 dhucs-interword 패키지를 로드했기 때문에 선언한 것으로서 이에 관련된 문제는 97 페이지의 2.2 절에 설명되어 있다.

이 내용을 firsttest.tex이라는 이름으로 저장한다.

¹hangul-ucs 4.0 이전 버전에서는 dhucs-ucshyper 라는 부가 패키지가 제공되었으나, hyperref의 기능 향상에 따라 불필요해졌으므로 ko_{TeX}에서는 이 부가 패키지를 제공하지 않는다.

```

1 \documentclass{article}
2
3 \usepackage[hangul,nonfrench,finemath]{kotex}
4 \usepackage[default]{dhucs-interword}
5 \usehangulfontspec{ut}
6 \usepackage[hangul]{dhucs-setspace}
7 \usepackage{dhucs-gremph}
8
9 \usepackage{ifpdf}
10 \ifpdf
11   \usepackage[unicode,pdftex,colorlinks]{hyperref}
12   \input glyphtounicode\pdfgentounicode=1
13 \else
14   \usepackage[unicode,dvipdfm,colorlinks]{hyperref}
15 \fi
16
17 \begin{document}
18
19 \section{들 어 가 기}
20 처음 문서를 작성합니다.
21
22 \end{document}

```

그림 4.1: 첫번째 문서, 예제

2.2 컴파일

이 문서를 컴파일하려면 명령행에서 다음과 같이 한다.

```
$ latex firsttest
```

PDF 출력을 얻으려면, `latex` 대신 `pdflatex`을 실행한다. 또는, `latex`으로 얻어진 `.dvi`에 대하여 `dvipdfmx`를 실행한다.

```
$ dvipdfmx firsttest
```

2.3 영문 글꼴

영문 글꼴은 일반적인 \LaTeX 문서의 경우와 마찬가지로 사용할 수 있다. `ko.TEX/utf`의 영문 글꼴 기본값은 OT1 인코딩의 CM이다. T1 인코딩으로 EC 글꼴을 사용하려면 다음을 `preamble`에 적어준다.

```
\usepackage[T1]{fontenc}
```

예를 들어 pxfonts 를 본문에서 쓰려면,

```
\usepackage{pxfonts}
```

와 같이 한다.

dhucs 자체가 font encoding 을 정의하고 있기 때문에, font encoding 을 바꾸는 명령, 예를 들면 `\usepackage[T1]{fontenc}` 과 같은 행은 dhucs 가 로드되기 전에 부르는 것이 좋다.

2.4 옵션들

ko_T_EX/utf 가 제공하는 옵션은 다음과 같다.

hangul 한글 문서 서식을 설정한다. 장과 절 포맷은 “제 1 장”과 같은 형태로 식자된다. 기본값은 한글 문서 서식을 설정하지 않는 것이다.

nojosa 자동조사 기능을 끈다. dhucs 초기 버전에서 다른 패키지와의 호환을 위하여 제공되었던 것이지만 현재는 굳이 이 옵션을 사용해야 할 필요가 현저히 줄었다. 그러나 옵션으로서는 유지되고 있다.

hanja 한글 문서 서식을 설정하는 것은 **hangul** 옵션과 같으나, 한글 이름 대신 한자 이름을 사용한다.

nonfrench 한글 문장에 대해서도 non-french-spacing 을 적용한다. 즉, 마침표 등의 뒤의 공백이 다른 공백보다 조금 더 길다. 기본값은 french-spacing 으로 식자하는 것이다.

finemath 원래는 문장 중의 수학 식과 이어지는 한글 사이에 적절한(fine) 간격을 부여하기 위한 것이었으나, 곧 문장부호 등을 포함한 개념으로 확대되었다.

strictcharcheck 한글 폰트 가운데 .tfm 은 있으나 해당 문자가 결락되어 있는 것이 있다. 이럴 경우에 에러를 보여주도록 하는 옵션이다. 이 옵션을 지시하지 않으면 문자가 없어서 식자하지 못한 것을 .log 에 기록만 하고 에러는 발생시키지 않는다.

2.5 부가 패키지들

ko_T_EX/utf 과 함께 제공되는 부가(contrib) 패키지들은 다음과 같다.

dhucs-cmap pdf_T_EX 을 이용하여 한글 문서를 작성할 때 한글 텍스트의 검색과 추출을 가능하게 하는 패키지이다. 주로 트루타입 글꼴을 사용할 때 필요하다.

`dhucs-interword` 한글 문서의 자간과 단어간격을 설정한다. `default` 옵션과 `HWP` 옵션을 쓸 수 있다.

`dhucs-setspace` 한글 문서의 행간을 설정한다. `hangul` 옵션을 반드시 지정해야 한다.

`dhucs-gremph \itshape`에 해당하는 한글 문자를 그래픽 글꼴로 바꾼다. 그래픽 대신 굵은 글꼴로 대신하는 `bfemph` 옵션을 쓸 수 있다.

`dhucsfm` 한글 라텍의 `hangulfn`을 `dhucs`로 포팅한 것으로, 한글 문서에 알맞은 여러 가지 각주짜기 방법을 제공한다.

`dhucs-trivcj` 간단한 일본어 또는 중국어 문단을 식자할 수 있게 하는 부가 패키지이다.

`dhucs-midkor` 1933년 이전 “옛한글”을 식자하기 위한 패키지이다.

`dhucs-enumitem` `enumitem` 패키지에 한글식 글머리 정의를 추가하는 작은 패키지이다. `enumitem` 패키지를 로드한 후에 불러쓰면 된다.

`dhucs-enumerate` `enumerate` 패키지에 한글식 글머리 정의를 추가하는 작은 패키지이다.

`dhucs-paralist` `dhucs-enumerate` 패키지와 같으나 `paralist` 패키지를 사용할 때 발생하는 문제를 해결하기 위한 것이다.

제 3 절 글꼴 사용과 선택

`ko.TEX/utf`에서 사용할 수 있는 기본 글꼴은 ‘은 글꼴 type 1’이다. 이에 대해서는 27 페이지 3 절에서 이미 상세히 밝혀두었다. 여기서는 이 기본 글꼴을 문서에서 사용자가 사용하는 방법에 대해서 기술한다.

우리는 글꼴 문제에 있어서 상대적으로 보수적인 태도를 취하려 한다. 즉, 하나의 문서에 기본적으로 사용되는 글꼴은 `rm`, `sf`, `tt` 세 가지라고 보는 것이다. 다른 아무런 설정 없이 기본값으로 문서를 작성하면, `rm`은 은 바탕, `sf`는 은 돋움, `tt`는 은 타자로 설정되어 있다. 이 이상의 글꼴을 하나의 문서에서 사용하는 것을 권장하지는 않지만 불가능하지도 않다. 필요하다면 은 글꼴 이외의 글꼴을 사용하는 것도 가능하다.

표 4.1: 글꼴 사용 방법

글꼴 가족	rm	sf	tt	emph
한글	utbt	utgt	uttz	utgr
한자	utbt	utgt	uttz	utgt

3.1 기본 글꼴을 변경하는 방법

ko.TeX/utf의 기본 글꼴 설정을 요약하면 표 4.1과 같다.

여기서 `emph`란, `dhucs-gremph` 패키지를 불러들 경우 이탤릭과 슬랜티드에 대응하는 한글 글꼴을 가리키는 것이다. 이 패키지를 로드하지 않은 상태에서는 `emph(itshape)`에 기울임 글꼴이 대응된다.² 강조의 방법으로서 글꼴 치환과 기울임 방법에 대한 논란은 이 글 뒷편에서 더 상세히 다룬다.

ko.TeX/utf는 모두 세 영역의 폰트를 사용하여 문서를 식자한다.

1. 영문자 영역. ASCII 문자들은 모두 여기에 들어간다. 즉 로마자 알파벳과 숫자, 일부 문장부호들이다. 이것은 영문 문서에서와 동일한 영문자 폰트를 이용하여 식자한다. 다른 설정이 없으면 OT1 인코딩의 CM이 기본값이며 T1 인코딩을 호출하면 `cm-super` 폰트를 EC 글꼴 문자로 사용한다. 만약 `lmodern`을 이용하고 싶으면 `\usepackage{lmodern}`이라고 하면 된다.
2. ‘한글’ 영역이란 [U+AC00]에서 [U+D7FF]까지 한글 음절문자 영역의 글자와 한글 자모 및 한글 호환자모 글자들에 해당한다. 지정된 한글 폰트로 식자한다.
3. ‘한자’ 영역이란 위의 영문자 영역과 한글 영역을 제외한 모든 문자가 포함된다. 한중일 호환 문장부호, 한자는 물론이고 일부 폰트에서 발견할 수 있는 PUA(사용자 영역) 옛한글 문자도 여기에 속한다.

이 세 영역은 완전히 분리되어 있고 서로 영향을 미치지 않는다. 즉 한글 영역의 글꼴을 바꾸더라도 영문자나 한자 영역의 글꼴의 선택이 바뀌지 않는다는 것이다. 다만 `rm`, `sf`, `tt`라는 세 글꼴군에 한하여 각각 해당되는 한글 글꼴과 한자 글꼴을 가져다가 식자한다. 예를 들어 `\sffamily` 명령이 불리면 영문자와 숫자는 `\sfdefault`인 `cmss` 폰트가 사용될 것이며, 한글은 이미 정의되어 있는 `sfhangulfont`, 여기서는 `utgt`가 식자될 것이다.

²다만 `oblivoir` 클래스에서는 `gremph`가 기본 설정이다.

글꼴 가족의 호칭은 대부분 네 개의 알파벳 문자로 표현된다. 기본 글꼴은 각각 `ut` 라는 폰트 제작자 부호와 `bt` 라는 폰트 가족 대표호칭으로 이루어져 있는데, `utbt` 라는 이름으로 호출되는 실제의 폰트들은 수백 개의 서브폰트들로 이루어져 있다. 문서 작성자는 실제 폰트 자체의 이름에는 신경 쓸 필요가 없고 `utbt` 또는 `utgt` 라는 호출 가족 명칭만 기억하면 될 것이다.

기본 글꼴 설정을 변경하려면 다음과 같이 한다. 여기 예시된 `XXaa` 등은 임의의 폰트 명칭인데, 이런 형식의 폰트 세트가 미리 시스템에 설치되어야 할 것은 당연하다.

```
\SetHangulFonts{XXaa}{XXbb}{XXcc}
\SetHanjaFonts{XXba}{XXbc}{XXca}
```

위의 명령은 각각 한글 글꼴과 한자 글꼴을 지정한다. 인자(parameter)는 세 개인데 각각 `rm`, `sf`, `tt` 글꼴에 대응한다. 예를 들어 본문 서체(바탕 또는 명조)만을 한겨레결체로 바꾸고자 하는 경우, (한겨레결체가 `hkbt` 라는 이름으로 시스템에 이미 설치되어 있다고 하고) 다음 명령을 지정하면 된다.

```
\SetHangulFonts{hkbt}{utgt}{uttz}
```

한겨레결체는 한자 영역이 없는 글꼴이므로 한자 폰트로 지정할 수는 없을 것이다.

문서 전체의 기본 글꼴을 변경하면서 폰트에 특정한 미세 타이포그래피 조절값을 적용하려면, 5.2 절에서 다루고 있는 대로 `hfontspec`을 이용하는 방법을 쓰는 것이 좋다. 새로운 폰트를 설치할 때 그 폰트에 해당하는 `hfontspec` 파일이 있는지 살펴본 후, 이것을 다음처럼 호출하는 것으로 충분하다.

```
\usehfontspec{XX}
```

물론 이 방법은 `hfontspec.XX`가 잘 설정되어 시스템에 설치되어 있어야 할 것이다.

`oblivoir` 클래스의 부수 스타일인 `hfontsel` 패키지를 이용하여 다음처럼 하는 방법도 있다.

```
\usepackage{hfontsel}
\SelectHfonts{XXaa,*,*}{*}
```

본문의 바탕(명조) 서체만을 `XXaa`로 바꾼다. 한자 글꼴을 별도로 지정해야 한다면, 예컨대

```
\SelectHfonts{hkbt,*,*}{utgt,*,*}
```

와 같은 방법이 있다.

본문에서는 다음과 같이 글꼴을 바꾼다. 이렇게 이루어진 지정은 현재의 구역 안에서³ 유효하다.

```
\SetSerifFonts{utbt}{utbt}  
\SetSansFonts{utgt}{utgt}  
\SetMonoFonts{uttz}{uttz}
```

두 인자는 각각 한글 영역 글꼴과 한자 영역 글꼴에 해당한다.

다음 단락은 `sanshangul`에 `utgs`를 할당하고 한자는 ‘은 옛글’로 식자한 예이다.⁴ 이 단락을 시작하기 전에 다음과 같이 선언하였다.

```
\SetSansFonts{utgs}{utyty}\sffamily
```

\TeX 은 스탠퍼드 대학의 도널드 크누쓰(D. Knuth) 교수가 만든 文書組版言語이다. \LaTeX 은 문서準備시스템이다.

문서의 일부에 대해서 폰트를 바꿀 때는 이 방법을 사용하기를 권장한다. 그래서 `\grfamily`, `\gtfamily` 또는 `\hfontfamily` 등의 `ko \TeX /euc` 명령은 제공하지 않는다.

폰트의 남용은 한글 문서의 중요한 폐해이다. 이를 억제하게 하기 위해서는, 기본 폰트들은 아주 쉽게 사용하게 하고, 다른 글꼴은 변경을 가능하게는 하되, 되도록 쉽지 않게 만드는 것이 옳다고 생각한다.

3.2 일시적으로 폰트를 변경하는 방법

기본 세 글꼴 이외의 별도 글꼴을 일시적으로 사용하려 하는 경우를 생각해보자. 일시적으로 글꼴을 바꾸는 데 사용되는 명령으로 `\SetAdhocFonts`가 제공된다. 이 명령도 두 개의 인자를 취하는데, 하나는 한글 글꼴이름이고, 두번째 것은 한자 글꼴이름이다. 앞서 언급한 대로 이 매크로에 의하여 한글과 한자 글꼴이 바뀌더라도 영문 폰트에는 영향을 끼치지 않는다.

우선 사용하려는 글꼴이 은 글꼴의 일부라면, 간단히 다음과 같이 새로운 명령을 자신이 정의해서 활용하도록 한다. 여기서는 은 필기 서체를 사용하는 경우를 생각해보자.

```
\newcommand\MYFNT{\ttfamily\SetAdhocFonts{utpg}{utgt}}
```

³`\begingroup`(또는 여는 중괄호)에서 `\endgroup`(또는 닫는 중괄호)까지

⁴이 부분이 에러없이 컴파일되려면 `extra` 폰트까지 설치되어 있어야 한다.

이와 같이 설정한 다음, 본문에서 다음과 같이 사용한다.

`{\MYFNT 여기에 단락이 옴}`

위의 예에서 중괄호로 이 단락을 묶어준 이유는 `\MYFNT` 명령의 범위를 한정하기 위해서이다. 중괄호로 닫은 이후에는 다시 원래의 폰트 설정으로 복귀한다. 중괄호가 혼란스러우면 명령형 명령을 정의하여 사용한다.

명령형 `\textMYFNT`를 다음과 같이 정의할 수 있다.

```
\DeclareTextFontCommand{\textMYFNT}{\ttfamily\SetAdhocFonts{utpg}{utgt}}
```

이 명령을 사용하여, `\textMYFNT{여기에 단락이 옴}`과 같이 코딩한 결과는 그림 4.2와 같다.

여기에 단락이 옴

그림 4.2: 폰트 지정 예제

이 명령의 첫째줄 `\ttfamily`를 먼저 선언한 이유는 영문 글꼴을 `tt family`로 하기 위해서이다. 이 선언은 한글에도 영향을 미치므로(앞서 설명하였다), 한글 글꼴을 바꾸기 전에 선언해둔다.

3.3 사용자 폰트를 활용하는 방법

만약, 원하는 글꼴이 은 글꼴이 아닌 경우라면 어찌할 것인가? 사용자 자신의 트루타입 한글 폰트를 `ko.TEX/utf`에서 활용하려 한다면, 먼저 해당 폰트를 사용하기 위해 필요한 파일 세트가 KTUG 등에서 제공되지 않는지 확인해보는 것이 필요하다. 예를 들면 백묵 글꼴, 문화부 글꼴, 윈도 기본 바탕·굴림 글꼴, 한겨레결체, 아리따 글꼴 등에 대한 `.tfm` 및 부수파일들이 별도로 제공되고 있다. 이 이외의 폰트를 활용하는 것은, 호환성이나 폰트의 라이선스 등의 문제로 그다지 권장할 만한 일이라고 하기 어렵다고 생각한다.

그러나 정말로 새로운 트루타입 폰트를 사용하려 하며, 폰트 라이선스 등에 대하여 자신이 책임질 수 있고, 약간의 복잡한 설정 작업을 기꺼이 해내려 한다면, `ttf2kotexfont` 유틸리티를 사용하면 된다. 이 유틸리티의 사용법은 이 문서 말미에 별도의 장으로 마련되어 있다(113 페이지, 9장). 한편, 일반적인 트루타입 폰트의 사용방법에 대해서는 “한글 레이텍에서 임의의 트루타입 사용하기”(김강수 & 주철, 2004:2007)라는 문서를 참고하라.

3.4 엄격한 문자 체크

한글 문서 작성 상황은 매우 복잡해서, 현재 사용중인 폰트에 사용자가 표현하고자 하는 모든 문자가 구비되어 있지 않은 경우가 많다. 기본 글꼴인 은 바탕만 하더라도 예컨대 한자는 KS X 1001에 해당하는 4888자밖에는 들어 있지 않다.

그런데 \TeX 은 .tfm이 아예 없다면 에러를 내지만 .tfm은 있으나 그 .tfm의 문자 정의에 해당 글자가 없는 경우에는 다만 글자를 식자하지 않고 경고만을 보이고 지나가면서 .log에 기록하는 것에 그친다. 그 결과 .log를 세심하게 살펴보면 혹 부지불식간에 특정 문자를 식자하지 못하는 결과를 초래하게 된다.

이를 위하여 도입된 옵션이 [strictcharcheck]이다. 이 옵션을 활성화하면 $\epsilon\text{-TeX}$ 의 원시명령을 이용하여 만약 .tfm은 있는데 특정 문자가 없다면 에러를 발생시키면서 컴파일을 멈추게 한다. 문서 작성자는 이를 통해 현재의 상황을 보다 잘 인식하고 폰트를 교체하거나 임시 폰트를 지정하는 등의 조치를 취할 수 있을 것이다.

이 옵션을 지정하지 않으면 에러를 보여주지 않고 다만 .log에만 기록한다.

제 4 절 몇 가지 한글화

4.1 기호문자 처리

기호문자 중 일부에 대해서는 미리 식자방법을 정의해두었다. $\frac{1}{2}$, $\frac{1}{4}$, \cdots 등의 문자와 같은 몇 가지 한글 기호문자는 [hangul] 옵션을 준 경우에 식자할 수 있다.

babel 패키지 등을 사용하거나 하는 경우 한글 기호문자의 $\backslash\text{euro}(\text{\text{€}})$ 역시 \TeX 코드로 변환되므로, 예컨대 eurofont 패키지 또는 eurosym 패키지를 사용하여 식자하도록 하여야 할 것이다.

Windows 기본 한글 입력기에서 특수문자를 입력하는 방법은 ‘ㄱ’, ‘ㄴ’ 등의 키를 누른 다음 [한자] 키를 눌러서 선택하는 것이다. 이 방법으로 입력되는 특수문자의 예를 표 4.2에서 보았다.⁵ ‘ㅎ’ 행에는 그리스 문자가 할당되어 있다. 텍스트 그리스 문자는 현재 본문 기본 한자 글꼴에 있는 것으로 식자되는데, 만약 더 나은 텍스트 그리스 문자가 필요하다면 babel 패키지를 이용하여 입력하도록 한다. 은 바탕 글꼴로 식자하면 이러하다: $\text{\TeX}\nu\acute{\eta}$.

기호 문자의 직접 입력과 처리가 가능하기는 하지만, 일반적으로 이 기호 문자를 문서에서 사용하는 것을 권장하지는 않는다. 기호 문자를 써야 할 경우에는 알맞은 \LaTeX 의 기호 문자 명령으로 얻도록 하는 편이 나을 것이다.

⁵ $\backslash\text{DeclareUnicodeCharacter}$ 명령을 이용하여 ucs의 문자 설정을 변경하는 방법도 있다.

표 4.2: Windows 특수문자 입력방식에 의한 특수문자

4.2 한글 문서 서식

한글 문서는 외국어 문서와는 달리 몇 가지 보조적 서식을 가지고 있다. 장절 명령이 별도로 짜여져야 하고, 행간이나 단어간격도 적절하게 배치되어야 한다. 이를 위하여 이 패키지는 [hangul] 패키지 옵션을 제공한다. 이 옵션이 지시되면 한글 이름, 한글 장절표제 등을 활성화하고 행간을 한글에 알맞게 설정한다. 또한, 장절 명령과 관련해서 ko_{TeX}/euc와 동일한 \kscntformat 명령이 작동한다. 그리고 한글 이름과 관련해서 \ksnamedef 명령도 작동한다. 이 명령의 의미와 사용법에 대해서는 88 페이지의 2.7 절을 보라.

한글 이름에 관한 사항은 ko_{TeX}/euc와 ko_{TeX}/utf가 동일하다. 89 페이지 표 6.2를 참고하라.

4.3 장절명령

article 문서에서 한글 서식을 사용하면서 장절 표제의 “제 1 절”과 같은 양식을 쓰지 않으려면 다음과 같이 설정하는 것이 일반적이다. 이 매크로는 ko_{TeX}/utf에 [hangul] 옵션을 지정한 경우에만 쓸 수 있다.

```
\kscntformat{section}{}{}
```

장절 표제 형식을 설정하기 위해서 ko_{TeX}/utf와 함께 제공되는 dhucs-sectsty를 사용하는 방법도 있다. 이것은 Rowland MacDonnell 씨의 sectsty 패키지를 한글화한 것이다. 100 페이지의 4 절을 보라. sectsty 패키지에 없는 새로 추가된 옵션은 [ensec] 인데, 이 옵션을 주면 ko_{TeX}/utf에 [hangul] 옵션을 주었더라도 제 1 절과 같은 형식이 아니라 영문 문서에서와 같이 제와 절이 붙지 않는 형식으로 식자된다. 그밖의 사용법은 sectsty 패키지에서와 같다. 다음은 절 모양을 바꾸는 한 가지 보기이다.

```
\sectionfont{\nohang\sffamily\centering}
```

다음 소절을 이런 방식으로 바꾸어보았다.

4.4 강조

한글 문서에서 강조(드러냄)를 처리하는 방법은 몇 가지가 있다.

4.4.1 기울인 글꼴 강조

강조(emph) 선언 \em이나 명령 \emph는 일반적으로 이탤릭 글자로 식자하는 것이 영문문서의 관행이다. 그러나 우리 글자에는 이탤릭이 없어서, 종래에는 fake

slanted 글꼴을 사용하여 왔다. 그러나 fake slanted는 pdf \TeX 이 지원하지 않을 뿐 아니라, 설령 구현된다 하더라도 강조의 의미는 별로 드러나지 않는 장식적인 문자꼴이 되어버리고 만다.

4.4.2 드러냄표 강조

(한글 맞춤법, 1988)의 부록 「문장부호」에 드러냄표가 규정되어 있다. 이것은 과거 한글 세로쓰기 서적에서 사용되던 방점을 찍는 방법의 연장선상에 있는 것이다.

드러냄표 강조는 한글 라텍스에서 시작된 것으로, ko \TeX /euc와 ko \TeX /utf가 모두 지원한다. `\dotemph`가 드러냄표를 식자하는 명령인데, 이 차이를 간단히 요약하면 다음과 같다.

	ko \TeX /utf	ko \TeX /euc	비고
상점	<code>\dotemph</code>	<code>\dotemph \dotem^a</code>	^a 선언형
고리상점	<code>\circemph*</code>	<code>\circemph \circem^a</code>	*[hangul]
임의의 상점	<code>\useremph*</code>		

`\dotemph` 명령으로 점을 한글 글자 위에 찍을 수 있다. [hangul] 옵션을 지시하면 `\circemph`와 `\dotemph` 그리고 `\useremph`, 세 개의 명령을 사용할 수 있고, 이 가운데 `\useremph`는 사용자가 설정 가능하다. ko \TeX /euc에서와는 달리 선언형은 제공하지 아니한다.

한글의 본 이름은 훈민정음이다.

중요한 것은 왜 사느냐가 아니라 어떻게 사느냐 하는 문제이다.

*** 정의 드러냄표는 다음과 같은 방식으로 사용할 수 있다.

```
\renewcommand\useremphchar{\tiny★}
\setlength\useremphraisedim{10pt}
\useremph{사용자 정의 드러냄표}
```

4.4.3 글꼴 변경 강조

최근 들어서는 돋움체 글꼴을 사용하는 것이 하나의 관례가 되어 가고 있다. h \TeX p는 여기에 그래픽 글꼴을 사용하였는데, 이것도 시각적으로 두드러지기만 한다면 나쁘지 않은 선택이라고 생각한다.

이 기능을 사용하려면 dhucs-gremp 패키지 로드한다. 만약 emph에 해당하는 글꼴을 그래픽이 아닌 다른 글꼴로 바꾸고 싶다면 다음과 같이 한다.

```
\usepackage[gremphhangul=unbm,gremphhanja=unyt]{dhucs-gremph}
```

첫번째 인자는 한글, 두번째 인자는 한자 글꼴을 지정하는 것이다. 또, `emph`에 해당하는 문자를 굵은 글꼴로 하고 싶으면,

```
\usepackage[bfemph]{dhucs-gremph}
```

와 같이 `[bfemph]` 옵션을 지정한다.⁶

문서의 중간에서 기울임 방식과 글꼴 변경 `\emph` 방식을 바꿀 수도 있다. 각각 `\regremph` 명령과 `\ungremph`를 사용한다. `\regremph`를 지정한 경우에는,

테스트. *test.*

와 같이 식자되고, `\ungremph`를 설정한 경우에는,

테스트. *test.*

와 같이 식자된다.

4.4.4 밑줄 강조

한글 맞춤법에는 밑줄 강조에 대해서도 언급하고 있다. `\underline`으로 밑줄을 그을 수 있지만, `ulem` 패키지를 사용하면 더 나은 결과를 얻을 수 있다.

다음 보기에서 명사가 `\uline{아닌}` 것은?

다음 보기에서 명사가 아닌 것은?

`ulem` 패키지는 옵션없이 로드하면 모든 `\emph`를 밑줄 강조로 바꿔놓는다. 이를 억제하고 `\uline`을 썼을 때만 밑줄을 그으려 한다면

```
\usepackage[normalem]{ulem}
```

과 같이 하여야 한다. 또한, `ulem` 패키지를 `ko.TeX/euc`에서 바로 사용할 수는 없으므로 `ko.TeX/utf`에서만 `ulem`을 이용한 밑줄 방식을 써야 한다. `ko.TeX/euc`에 대해서는 99 페이지의 3절을 참고하라.

⁶은 글꼴 중에는 `boldface` 글꼴이 따로 마련되어 있지 않은 것이 많다는 점에 주의하라. 이 경우에는 약간의 `Warning`을 내면서 그냥 보통 글꼴로 식자된다.

4.5 자간, 행간

한글 타이포그래피의 가장 기본적인 요소인 자간은 5.2에서 소개한 한글 폰트 스펙의 일부로 포함하여 취급하게 되었다.

종래 자간·단어간격을 지원하던 `dhucs-interword` 패키지는 여전히 제공되는데, 자간만을 일시적으로 바꾸거나 할 경우 부분적으로 활용할 수 있을 것이다. 그러나 이제 자간의 변경은 폰트 스펙을 이용하는 바꾸는 것이 좋겠다. 이 패키지는 단어간격의 제어를 위해 사용하는 정도로 활용될 수 있을 것으로 본다. 행간을 조절하기 위해서는 `dhucs-setspace` 패키지를 사용한다.

한글 자간의 설정에 있어서 반드시 알아두어야 하는 사실을, `[finemath]` 옵션이 활성화된 경우에만 자간 설정이 의미를 갖는다는 것이다. 이 옵션을 주지 않은 경우에는 자간이 언제나 0pt이며 이것을 바꿀 수 없다. `dhucs-interword` 패키지를 로드하면 `finemath`가 활성화되어 있지 않을 경우 오직 단어간격에만 영향을 미친다.

이 패키지를 사용할 때는 순서에 주의하여야 하는데, `dhucs-interword` 패키지를 `dhucs-setspace` 패키지보다 먼저 올리도록 하는 것이 좋다.

```
\usepackage[hangul]{dhucs}
\usepackage[default]{dhucs-interword}
\usehangulfontspec{ut}
\usepackage[hangul]{dhucs-setspace}
```

`dhucs-interword` 패키지를 로드한 직후에 문서에서 사용되는 한글 폰트의 폰트 스펙을 활성화해주는 것은 좋은 선택이다(세번째 줄).

`dhucs-interword` 패키지에 대해서는 97 페이지 2.2 절, `dhucs-setspace` 패키지에 대해서는 98 페이지 2.3 절을 보라.

4.6 장평

특별한 효과를 위해서 “좁은 장평”을 사용하려 하는 경우가 있을 수 있다. 기본 글꼴에 `c-series`의 93% 장평⁷이 정의되어 있으므로 이것을 사용할 수 있다. 다음 문단은 좁은 장평으로 텍스트를 식자한 것이다. `\fontseries{c}\selectfont`를 선언하였다. 당연한 말이지만, 좁은 장평이 준비되어 있지 않은 글꼴—자모바탕 등—은 이렇게 하더라도 좁은 장평을 얻을 수 없다. 어떤 폰트에 좁은 장평이 마련되어 있는지에 대해서는 27 페이지 3 절에서 설명하였다.

⁷표준 NFSS에서 `c-series`는 보통 75%의 좁은 폭이다. 그러나 `kaTeX`에서는 92 또는 93%를 `c-series`에 할당하였다. 한글의 75% 장평은 거의 무의미한 변형이라고 생각한다.

아아, 나는 이제야 도(道)를 알았도다. 마음이 어두운 자는 이목이 누(累)가 되지 않는다. 이목만을 믿는 자는 보고 듣는 것이 더욱 밝혀져서 병이 되는 것이다. 이제 내 마부가 발을 말굽에 밟혀서 뒷차에 실리었으므로, 나는 드디어 혼자 고삐를 늦추어 강에 띄우고, 무릎을 구부려 발을 모으고 안장 위에 앉았다. 한번 떨어지면 강이나 물로 땅을 삼고, 물로 옷을 삼으며, 물로 몸을 삼고, 물로 성정을 삼을 것이다. 이제야 내 마음은 한번 떨어질 것을 판단한 터이므로, 내 귓속에 강물 소리가 없어졌다. 무릇 아홉 번 건너는데도 걱정이 없어 의자 위에서 좌와(坐臥) 하고 기거(起居) 하는 것 같았다.

장평을 자유로이 바꾸는 것은 허용되지 않는다.⁸ 다만 미세 타이포그래피를 위하여 아주 미세한 장평의 조절을 pdf_T_EX이 행하는 경우가 있는데 이것은 c-series를 선택하는 것과는 다른 문제이다.

4.7 nonfrench spacing

nonfrench spacing이란, 마침표 뒤에 별도의 공백을 조금 더 주는 조판 양식을 말한다. 문장의 마침표를 더 잘 드러나게 할 목적으로 사용하는 공백이라고 한다. 유럽에서는 이 추가 공백을 허용하지 않는 것이 일반적인 조판 형식인 듯하다.

이 기능을 활성화하려면 [nonfrench] 옵션을 지정한다. 추가 공백의 기본 크기가 마음에 들지 않으면 \xspaceskip으로 변경할 수 있다.

다음 두 단락은 동일한 문장을 각각 french-spacing과 nonfrench-spacing으로 식자한 예이다. 이 인용문은 연암의 “하룻밤에 아홉 번 물을 건너다”(박지원, 1780)로부터 왔다.

Frenchspacing:

아아, 나는 이제야 도(道)를 알았도다. 마음이 어두운 자는 이목이 누(累)가 되지 않는다. 이목만을 믿는 자는 보고 듣는 것이 더욱 밝혀져서 병이 되는 것이다. 이제 내 마부가 발을 말굽에 밟혀서 뒷차에 실리었으므로, 나는 드디어 혼자 고삐를 늦추어 강에 띄우고, 무릎을 구부려 발을 모으고 안장 위에 앉았다. 한번 떨어지면 강이나 물로 땅을 삼고, 물로 옷을 삼으며, 물로 몸을 삼고, 물로 성정을 삼을 것이다. 이제야 내 마음은 한번 떨어질 것을 판단한 터이므로, 내 귓속에 강물 소리가 없어졌다. 무릇 아홉 번 건너는데도 걱정이 없어 의자 위에서 좌와(坐臥) 하고 기거(起居) 하는 것 같았다.

⁸그러나 사용자가 꼭 필요하다면 주어진 폰트로부터 특정 장평의 tfm을 얻어내고 이것을 사용하도록 하는 방법이 있다. 다만 그 절차가 좀 복잡할 따름이다. 이 방법을 설명하는 것은 이 문서의 목적과 무관하다.

Nonfrenchspacing:

아아, 나는 이제야 도(道)를 알았도다. 마음이 어두운 자는 이목이 누(累)가 되지 않는다. 이목만을 믿는 자는 보고 듣는 것이 더욱 밝혀져서 병이 되는 것이다. 이제 내 마부가 발을 말굽에 밟혀서 뒷차에 실렸으므로, 나는 드디어 혼자 고삐를 늦추어 강에 띄우고, 무릎을 구부려 발을 모으고 안장 위에 앉았다. 한번 떨어지면 강이나 물로 땅을 삼고, 물로 옷을 삼으며, 물로 몸을 삼고, 물로 성정을 삼을 것이다. 이제야 내 마음은 한번 떨어질 것을 판단한 터이므로, 내 귓속에 강물 소리가 없어졌다. 무릇 아홉 번 건너는데도 걱정이 없어 의자 위에서 좌와(坐臥)하고 기거(起居)하는 것 같았다.

4.8 각주

한글 라텍은 한글 문서의 각주 판짜기를 위한 `hangulfn.sty`를 제공한다. 이 각주 판짜기 스타일의 `koTeX/utf` 버전은 `dhucsfn` 패키지이다. 둘 사이의 다른 점은 `dhucsfn`이 영문 옵션만을 받아들인다는 점이다. 기본값은 `superscript`, `hang`이다.

첨자	<code>superscript</code>
괄호	<code>parenthesis</code>
내어쓰기	<code>hang</code>
다항이어쓰기	<code>multipara</code>
단순이어쓰기	<code>para</code>
왼쪽맞춤	<code>leftflush</code>
들여쓰기	<code>indent</code>
들여왼쪽맞춤	<code>leftflushindent</code>
들여내어쓰기	<code>hangpar</code>
들여괄호맞춤	<code>varhangpar</code>

이 패키지의 사용법과 각각의 각주 모양 예시는 93 페이지 1 절을 보라.

4.9 한글식 카운터

그림 4.3과 같은 한글 카운터 형식을 제공한다. `[hangul]` 옵션을 주면 다음과 같은 카운터 형식을 몇 개 더 사용할 수 있다.

```
\hNum{cnt}, \hanjanum{cnt}, \HArabic{cnt}
```

<code>\newcounter{test}</code>	<code>\setcounter{test}{2}</code>
<code>\jaso{test}</code>	ㄴ
<code>\pjaso{test}</code>	(ㄴ)
<code>\ojaso{test}</code>	㉞
<code>\gana{test}</code>	나
<code>\ogana{test}</code>	㉞
<code>\pgana{test}</code>	(나)
<code>\onum{test}</code>	②
<code>\pnum{test}</code>	(2)
<code>\oeng{test}</code>	㉠
<code>\peng{test}</code>	(b)
<code>\hnum{test}</code>	둘
<code>\Hnum{test}</code>	둘째
<code>\hroman{test}</code>	ii
<code>\hRoman{test}</code>	II

그림 4.3: 한글식 카운터

`cnt` 값이 2일 때, 각각 이, 二, 2로 표현된다.

4.10 자동조사 명령

자동조사는 주로 사후적으로 결정되는 숫자 및 문자열(예컨대 `\ref`의 결과 확정되는 참조번호)에 붙는 조사의 형태교체를 자동화해주는 기능을 말한다. 참조된 숫자가 1일 경우에 목적격조사는 ‘을’이 붙어야 하지만 2일 경우에는 ‘를’이 붙어야 한다. 이 참조된 숫자들은 사후적으로 확정되므로 미리 ‘을’ 또는 ‘를’ 가운데 어떤 것을 써야 할지 알 수 없기 때문에, 우리말 문서의 작성에서는 빼놓을 수 없는 기능이다.

자동조사 명령은 다음 열두 가지가 있다.⁹

`\이` `\가`, `\을` `\를`, `\와` `\과`, `\로` `\으로`, `\은` `\는`, `\타` `\이타`

마지막의 `\타`, `\이타`는 서술격조사 어간 ‘이’에 어미(의 일부)가 붙은 형태로서 서술격조사 어간이 생략되는 경우가 이 이외에도 더 있지만, 다른 경우에는 명백히 생략되지 않아도 문장이 어색하지 않다고 보아 ‘(이)라’만을 자동조사 명령으로 채택한 것이다. 자동조사의 기능과 그 사용법은 85 페이지 2.6 절을 참조하라.

⁹koTeX/euc에는 ‘_’와 ‘\|’가 더 있다.

자동조사 사용에 있어서 알아두어야 할 점이 하나 있다. `\section` 등 장절명령이나 `\caption`과 같은 명령의 인자로 들어간(소위 “moving arguments”) 자동조사 명령은 `\ref` 등이 `\protect`되어 있다면 대체로 잘 작동한다. 그러나 `\nameref` 되는 문자열 안에 들어 있는 자동조사 명령은 `hyperref` 패키지의 문제점으로 인하여 오류를 토해내므로 주의를 요한다.

4.11 enumerate의 항목 머리

`enumerate` 환경에서 글머리로 한글식 카운터를 쓰고자 할 때가 있다. 보통의 방법은 다음과 같이 하는 것이다.

```
\renewcommand\theenumi{\pgana{enumi}}
\renewcommand\labelenumi{\theenumi}
\begin{enumerate}
\item 가나다
\item 라마바
\item 사자차
\item 카파하
\item 카타과
\end{enumerate}
```

결과는 다음과 같다.

- (가) 가나다
- (나) 라마바
- (다) 사자차
- (라) 카파하
- (마) 카타과

이 코딩은 상당히 번거롭고, 일단 카운터 형식이 바뀐 후로 계속 영향을 미친다는 문제가 있다.

한글식 카운터 형식을 글머리에 붙이는 쉬운 방법으로 두 가지가 있다. 하나는 `enumerate` 패키지의 옵션 정의를 이용하는 방법인데, 원래 이 패키지가 제공하던 기능을 한글식 글머리로 확장한 것이다.

```
\usepackage{dhucs-enumerate}
```

이 선언은 `enumerate` 패키지를 스스로 로드한다. 그리고 본문에서

```
\begin{enumerate}[항목 ㉔)]
\item 가나닥
\item 락마박
\end{enumerate}
```

과 같이 항목 머리의 첫 글자를 옵션으로 지시한다. 위의 결과는 다음과 같다.

항목 ㉔) 가나다

항목 ㉕) 라마바

이렇게 항목 머리로 지정할 수 있는 것은 원래의 `enumerate` 패키지가 제공하는 A, a, i, I, 1 이외에 추가로 정의된 가, ㄱ, ㉑, ㉒, (ㄱ), (㉑), ①, (1), (a), ②, i, I 등이 있다.

한편, 문단형 리스트를 구현하는 역할을 하는 `paralist` 패키지도 이와 비슷한 기능을 제공하는데, 이를 위하여 `dhucs-paralist` 패키지가 마련되어 있다.

`enumitem` 패키지를 이용하는 경우에는 먼저 `preamble`에 다음과 같이 선언한다.

```
%\usepackage{enumitem}
\usepackage{dhucs-enumitem}
```

그리고, 다음과 같이 코딩한다.

```
\begin{enumerate}[label={\bfseries\pgana*.}]
\item 가나닥
\item 락마박
\end{enumerate}
```

`enumitem` 패키지에는 이외에도 요긴한 기능이 많다. 예컨대 만약 문서 전체의 `enumerate`의 항목 머리를 전부 일정한 형식으로 바꾸고자 한다면 `enumitem` 패키지의 전역 형식 선언을 이용할 수 있다(ver2.0). 다음 예는 `label`과 더불어 항목 간 간격도 함께 없앤 예이다.

```
\setenumerate[1]{label={\pgana*},noitemsep}
\setenumerate[2]{label={\ogana*},nolistsep}
```

`enumerate` 패키지와 `enumitem` 패키지는 서로 동일한 환경을 다른 방식으로 재정의하기 때문에 함께 쓰기는 어려울 것이다. 둘 가운데 적당한 것을 골라서 사용하면 되는데, 어느 경우든 한글식 항목 머리를 정의할 수 있게 되었다.

제 5 절 미세 조정

ko \TeX /utf의 가장 큰 특징 중 하나는 한글 문서 전반에 걸쳐 미세 조정을 적용하여, 미세 타이포그래피를 부분적으로 구현하려 하였다는 것이다. 이 기능은 [finemath] 옵션에 의하여 모두 활성화된다.

5.1 수식과의 간격

\TeX 의 탁월한 점 가운데 하나가 수식 조판일 것이다. 특히 \TeX 수식은 미세한 간격을 적절하게 배분함으로써 아름다운 수식을 만들어내는 것으로 유명하다. 영문의 경우 행중 수식은 항상 앞뒤에 스페이스를 두면서 다음 단어와 연결되므로 문장 중에 들어가도 어색하지 않은 데 비해, 한글 문서에서는 조사 등이 수식 뒤에 잇대어 붙기 때문에 이 사이에 간격이 없으면 수식 폰트와 한글 폰트 사이에 약간 어색한 상충이 생긴다.

이 문제를 해결하기 위한 다양한 연구와 토론이 KTUG에서 이루어졌다. 그 결과, [finemath] 라는 새로운 옵션이 생기게 되었다. [finemath] 옵션을 주면 대부분의 행중수식과의 간격을 자동으로 설정한다. 이 효과는 수식, 영문자, 괄호 등에도 영향을 미친다.

다음은 [finemath] 로 조판한 결과이다. 한글 문자와 이어지는 위치에 미세한 간격이 추가되고 있음을 확인해보라.

x 에 임의의 값이 주어지면 $f(x)$ 를 통해서 그 x 에 대응하는 y 를 계산할 수 있다. 예를 들어 $x=2$ 로 x 값이 주어졌다면 $x+y$ 는....
숫자 01234567과 영문(알파벳)자모 abcd와 괄호에도 영향을 미친다.

5.2 한글 폰트스펙

한글 문서의 조판에서 영문 폰트와의 불일치 또는 부조화가 오래도록 문제가 되어왔다. 이 문제를 근원적으로 해결하려면 한글 폰트의 사용방법과 디자인 전반에 걸친 재검토가 요구된다.

궁극적으로 만족할 만한 해결책을 얻기까지는 더 많은 연구와 노력, 폰트 디자인 등이 필요할 것으로 생각하나, 우선 이 문제를 피해가면서 아름다운 한글 문서를 조판하기 위하여 ko \TeX /utf는 한글 폰트스펙이라는 개념을 도입하였다.¹⁰

한글 폰트스펙은 hfontspec.?? 라는 별도의 파일로 지정한다. 이 파일에 적어 넣을 수 있는 스펙값들은 대략 다음과 같다.

¹⁰ 폰트 스펙은 finemath 옵션을 선언해야 활성화된다.

- hangul unit 값 (hu)
- 한글 기본 자간 (interhchar)
- 문장부호 수직위치 조정값
- 한글 및 한자 폰트 패밀리 명칭

예를 들면, `hfontspec.ut` 이라는 기본 글꼴 폰트 스펙의 내용은 다음과 같이 되어 있다.

```
hu = .057175em
interhchar = -.0852em
fullstoplower = .15ex
exclamationlower = .2ex
questionlower = .2ex
serifhangulfont = utbt
sanshangulfont = utgt
monohangulfont = uttz
serifhanjafont = utbt
sanshanjafont = utgt
monohanjafont = uttz
```

위의 모든 값이 지정되어야 한다.

hu (hangul unit) 이라는 단위는 ko_TeX 이 내부적으로 사용하는 (논리적인) 길이 단위로서, 한글 1 전각 폭의 $1/16$ 로 정의된다. 여기서 1 전각이란 1em을 의미하는 것이 아니라 폰트 디자인상 평균적인 가로폭의 길이를 의미하는 것인데, 위의 기본 설정에서 이 값을 $.057175em$ 으로 잡은 것은 ko_TeX 의 기본 자간 $-0.0852em$ 을 제외한 $0.9148em$ 을 16등분한 값이다. 즉, $16 \times hu + interhchar = 1em$ 이 되도록 되어 있다. 이 길이 단위의 필요성과 사용에 대해서 (김강수, 2007b)에서 논의된 바 있다. hu는 `[finemath]`의 수평 간격에 영향을 미친다.

사용자가 이 폰트 스펙 값을 임의로 바꾸는 것은 판면에 심각한 영향을 미칠 수 있으므로, 자신이 무슨 일을 하고 있는가 확신이 있을 때만 변경하도록 하라. 폰트 스펙 값들은 폰트 패키지를 만드는 사람이 제공하는 것이 옳을 것이다.

위에 적은 값을 이용하여 자간을 설정하고 문장부호의 수직 위치를 조절하게 하려면, `[finemath]` 옵션을 활성화하면 된다. 문서 전체에 대해서 또는 일부에 대해서 문장부호 수직 위치조절이 적용되지 않게 하려면

```
\disablehangulfontspec
```

이라고 선언한다.

특정 폰트 스펙을 적용하도록 하는 매크로는 다음과 같다. 예를 들어 `xx` 라는 폰트에 대하여 `hfontspec.xx`가 준비되어 있다면,

```
\usehangulfontspec{xx}
```

를 선언하면 된다. 기본 글꼴(은 글꼴 type 1)에 대한 폰트 스펙(`ut`)은 `kotex`을 `\usepackage`하는 시점에서 자동으로 불러들인다.

아래 예문은 각각의 경우 문장부호 및 자간 등이 어떻게 달라지는가를 예시한 것이다. 문장 부호의 타이포그래피에 대해서는 (김강수, 2007b)에서 그 간의 토론을 요약하고 의견을 제시하였고 이것을 `ko.TEX`에서 구현한 것이다.

한글 문장부호에 영문 폰트의 것을 쓸 때는 베이스라인 불일치 등을 교정해주어야 한다. 예를 들어, 온점, 느낌표, 물음표 등의 수직위치가 그러하다. 자! 비교해볼까?	한글 문장부호에 영문 폰트의 것을 쓸 때는 베이스라인 불일치 등을 교정해주어야 한다. 예를 들어, 온점, 느낌표, 물음표 등의 수직위치가 그러하다. 자! 비교해볼까?
--	--

5.3 Micro Typography

`pdfTEX`의 저자인 Hàn Thế Thành의 박사학위논문 *Micro-typographic extensions to the T_EX typesetting system*(Thành, 2000)은 `pdfTEX`이라는 엔진으로 margin kerning과 font expansion을 가능하게 해주는 마이크로 타이포그래피 개념을 T_EX 세계에 도입하였다.

마진 커닝(margin kerning)이란, 여백 공간이 가지런히 보이도록 하기 위해 글자들을 텍스트 블록의 마진 쪽으로 아주 조금 이동시켜주는 테크닉이다. 마진 커닝을 적용하지 않으면 일부 문자들이 마진 경계가 시작하기 전에 끝나기 때문에 오히려 울퉁불퉁하게 보인다. 이것은 문장부호 끌어내기(hanging punctuation)와 비슷한 것이지만 문장부호만이 아니라 일반적인 글자에도 적용할 수 있다. 이를 적용함으로써 텍스트 블록의 모양을 현저히 개선할 수 있다.

폰트 확장(font expansion)이란 단어 간격이 좀더 균일하게 보이도록 하기 위해 폰트의 폭을 약간 좁게 혹은 넓게 만드는 테크닉이다. ... 폰트를 넓게 혹은 좁게 만듦으로써 개행 엔진이 더 나은 행나눔을 할 수 있게 된다.¹¹

¹¹Hàn Thế Thành, (Thành, 2001), p. 146.

나는 여름 방학이 되어 집
에 내려 오면 한 번씩은 이
집을 찾는다. 이 집에는 나
보다 한 살 아래인, 열세 살
되는 누이뻘 되는 소녀가 있
었다. 실상 춘수를 따져 가
며 통내외까지 할 절척도 아
니지만, 서로 가깝게 지내는
터수라, 내가 가면 여간 반
가워하지 아니했고, 으레 그
소녀를 오빠가 왔다고 불러
내어 인사를 시키곤 했다. 소
녀의 몸매며 옷매무새는 제
법 색시꼴이 박히어 가기
시작했다. 그때만 해도 시골
서 좀 범절있다는 가정에서
는 열 살만 되면 벌써 처녀
로서의 예모를 갖추었고 침

font expansion 적용 안함

나는 여름 방학이 되어 집
에 내려 오면 한 번씩은 이
집을 찾는다. 이 집에는 나
보다 한 살 아래인, 열세 살
되는 누이뻘 되는 소녀가 있
었다. 실상 춘수를 따져 가
며 통내외까지 할 절척도 아
니지만, 서로 가깝게 지내는
터수라, 내가 가면 여간 반
가워하지 아니했고, 으레 그
소녀를 오빠가 왔다고 불러
내어 인사를 시키곤 했다. 소
녀의 몸매며 옷매무새는 제
법 색시꼴이 박히어 가기
시작했다. 그때만 해도 시골서
좀 범절있다는 가정에서는
열 살만 되면 벌써 처녀로
서의 예모를 갖추었고 침선

font expansion 적용

그림 4.4: Font Expansion

한글 문서에 있어서 행나눔을 결정할 때 부득이하게 ‘병병한’ 공백을 남기고 개행
해야 하는 경우가 빈번했다. 또한 문장부호가 행 끝에 걸리는 경우 여백의 모양이 균
일하지 않게 보이는 것을 어찌할 수 없었다. 마진 커닝과 폰트 확장이 매우 요긴하게
요청되는 바였지만, 이를 쉽게 구현하기 어려웠으며, 간간이 마진 커닝 정도가 쓰이는
데 그쳤다. $\text{ko}\text{T}\text{E}\text{X}/\text{utf v0.1.0}$ 은 $\text{pdf}\text{T}\text{E}\text{X}$ 의 폰트 확장을 활용할 수 있게 한다.¹²

폰트 확장을 한글에 적용하기 위해서는 다음 조건이 갖추어져야 한다.

1. $\text{pdf}\text{T}\text{E}\text{X}$ 으로 컴파일할 것
2. type 1 혹은 트루타입 글꼴이 사용될 것
3. microtype 패키지에 의한 설정이 활성화될 것

그림 4.4는 폰트 확장을 적용하지 않은 경우와 그것을 적용한 경우의 행나눔과 스페
이스 간격을 비교해본 것이다. 더 균일하고 아름다운 판면이 생성되는 것을 눈으로
확인할 수 있을 것이다.

샘플 문서의 소스 중에서 font expansion과 관련된 부분은 다음과 같다.

```
\usepackage[verbose=true]{microtype}
```

¹² 김도현, “microtype 을 아시나요,” <http://kts.ktug.kr/node/207>.

면주와 페이지 스타일에 대한 부분이다. 일반적으로는 PageStyle에서 “running heading”이라 한 것이다. 이것을 정의하는 데는 fancyhdr 패키지가 쓰인다. 이 패키지만으로 원하는 거의 모든 설정을 할 수 있기 때문에 여기서 면주에 대하여 더 자세한 세부사항은 생략하려 한다. FAQ (FancyHdr) 페이지를 참고하라.

그림 4.5: 행나눔 문제

면주와 페이지 스타일에 대한 부분이다. 일반적으로는 PageStyle에서 “running heading”이라 한 것이다. 이것을 정의하는 데는 fancyhdr 패키지가 쓰인다. 이 패키지만으로 원하는 거의 모든 설정을 할 수 있기 때문에 여기서 면주에 대하여 더 자세한 세부사항은 생략하려 한다. FAQ (FancyHdr) 페이지를 참고하라.

그림 4.6: dhucs의 행자름 처리

```
\DeclareMicrotypeSet{dhucsmicro}{encoding=LUC}
\UseMicrotypeSet[expansion]{dhucsmicro}
```

제 6 절 행나눔

한글에는 하이픈이 없고 일부 문장부호의 금칙처리를 제외하고는 원칙적으로 모든 문자 앞이나 뒤에서 행을 나눌 수 있다.

6.1 영문자와 숫자 뒤의 한글 개행

그림 4.5에서 표시한 곳은 원본에 영문과 조사가 붙어 있는 부분이다. 빨간색 동그라미 부분에서 영문 단어가 overfull이 발생했음에도 불구하고 뒤에 붙는 한글 조사와 분리되지 않아서 “예”까지 오른쪽으로 빠져나가 있다. 종래 한글 라텍스에서는 이러한 문제가 자주 발생하여 문서작성자를 괴롭히곤 했다.

ko₂TeX은 개선된 행나눔 기능이 작동한다. 그림 4.6은 ko₂TeX을 사용했을 때 이 부분이 어떻게 처리되는가 보여준다.

6.2 여는 괄호 앞의 행자름

여는 괄호와 관련해서 다음을 지적해둔다. 현재 여는 괄호는 영문자 뒤에 직접 이어져 나오지 않는 한 새로운 줄을 시작할 수 있다. 신정식 님께서 알려주신 바에 의하면¹³

¹³<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=4825>

면주(面柱)는 흔히 PageStyle에서 “running heading”이라고 불리는 (호칭되는) 것이다. 이것을 정의하는 데는 fancyhdr 패키지가 쓰인다. 이 패키지만으로 원하는 거의 모든 설정을 할 수 있기 때문에 여기서 면주에 대하여 더 자세한 세부사항은 생략하려한다. FAQ(FancyHdr) 페이지를 참고하라.

한 가지만 지적하자면, [여덟글자를써넣고] L^AT_EX은 “고아와 과부 (orphans and widows)”라는 재미있는 이름으로 불리는 외따로 떨어진 줄을 허용하지 않는 훌륭한 기능이 있어서 예컨대 한 줄만이

그림 4.7: 여는 괄호의 행자름

모든 종류의 여는 괄호는 새로운 줄을 시작하는 것이 유니코드 표준이라고 한다. 그러나 영문으로 글을 쓸 때는 여는 괄호 앞에 하나의 공백을 두게 되어 자연스럽게 새로운 줄을 시작할 수 있을 것이므로 영문자 사이의 여는 괄호를 개행가능하게 하여야 할 필요를 발견하지 못하였다. 그러나 한글과 영문이 만나는 곳에서는 행나눔이 가능하여야 한다. 그림 4.7을 보라.

만약 이 그림에서 “FAQ(FancyHdr)”의 괄호 사이에서 행을 잘라야 한다면 여는 괄호 앞에 공백을 두어 FAQ(FancyHdr)와 같이 써야 할 것이다.

이밖에, ko_TE_X은 한글 단어 뒤에 괄호 안에 영문을 병기하는 경우 이 영문이 행 끝에 걸리더라도 영문 하이픈처리가 작동하도록 하고 있다.

6.3 이른바 “고아” 문제

“고아”란 한 문단의 마지막 행이 한 글자와 종지부호로만 끝나는 경우를 일컫는다. 한 행 전체 길이에 비교해보았을 때 한 글자 한 행은 그 비중이 너무 작아서 때때로 글읽기의 효율을 방해하고 불필요한 공백이 생겨나게 하여 판면의 색조를 열게 만드는 것으로 생각되어, 일찍부터 한글 조판의 금기 사항 중 하나였다.¹⁴

ko_TE_X에서는 고아 회피가 부분적으로 자동화되었다. 원칙적으로 문단 마지막에 아무런 추가 코드 없이 \par 또는 문단 종지 문자가 있을 때는 고아 앞에서 \nobreak가 작동하는 것과 같은 효과를 가진다. 그러나 문단 끝에 추가적인 코드나 문자가 더 있을 때 때때로 이 자동 고아 회피가 동작하지 않는 경우도 있으므로, 이럴 때는

¹⁴영어권에서 ‘고아와 과부’라고 할 때는 이와 조금 다른 의미로 쓰이는 경우도 있다. 예컨대 Peter Wilson의 (Wilson, 2006), p. 29에는 “고아란 한 페이지의 마지막에 한 줄 또는 두 줄이 남는 것을 말한다. Brighthurst의 암기를 위한 트릭에 따르면, ‘고아는 미래만이 있고 과거가 없는 것’이다.”라고 쓰고 있다. 즉, 과부와 고아가 모두 행나눔에 관한 문제가 아니라 페이지 나눔에 관한 문제로 취급되고 있는 것이다. 그러나 이 글에서는 우리나라의 관행적 명칭인 ‘문단 끝 한 글자 한 행’을 (편의상) 고아라고 부르기로 한다.

`\nobreak`를 사용자가 직접 지정하여야 한다. 특히 문단의 마지막 글자가 “다.”일 경우에 한하여 사용자가 고아 회피를 강제하도록 “\다” 매크로가 새로 도입되었다.¹⁵ 각 주 안과 같이 문단 종지 문자 없이 문단이 종료됨으로써 자동 고아 회피가 동작하지 않는 경우에 이 매크로를 적절하게 활용할 수 있다.

비록 모든 경우에 완전한 고아 회피가 적용되는 것은 아니지만, 이만큼이라도 자동화됨으로써 한글 문서 조판의 관행적 규약에 더 가까이 접근했다 하겠다.

제 7 절 한글 PDF 만들기

24 페이지 2 절에서 pdf 제작에 관한 사항을 개괄하였다. 여기서는 이 문제에 대하여 `ko.TeX/utf`의 관점에서 조금 더 다루려고 한다.

한글 pdf 문서를 만드는 데 있어서 지금까지 요청되어 왔던 것은 대략 다음과 같다.

- 한글 텍스트의 검색과 추출
- pdf bookmark에 한글 구현
- beamer 등 발표 문서 작성 패키지 지원

7.1 하이퍼링크

하이퍼링크를 구현하기 위해서는 `hyperref` 패키지를 이용한다. 이 패키지 사용에서 주의할 점은 한글 레이블이나 참조자를 사용하면 아니된다는 것이다.

`hyperref` 설정은 다음 세 가지 방법으로 이루어진다.

1. `\usepackage[...]{hyperref}`의 옵션
2. `\hypersetup` 문장
3. `hyperref.cfg`에 미리 설정된 옵션

예컨대 한글 문서를 주로 작성하는 경우 `hyperref`의 `[unicode]` 옵션이 필요하다. 그리고 클래스 자신이 `hyperref`을 스스로 로드하는 경우에는 (1)이나 (2)와 같은 방법으로 이 옵션을 활성화할 수 없게 되어 있는 경우가 있다. 이럴 때는 (3) 방법을 사용한다. 즉 `hyperref.cfg`에 (2)의 문장을 미리 적어넣어두는 것이다.

¹⁵<http://kts.ktug.kr/node/204>.

표 4.3: dvi to pdf drivers

driver	type1	truetype	ttf slanted	opentype	graphic
dvips	O	X	X	X	eps
pdf \TeX	O	O	X	X	pdf/png/mps/jpg
DVIPDFMx	O	O	O	O	(eps)/pdf/jpg/png/mps

7.2 폰트 문제

pdf를 제작하는 방법에 따라 사용할 수 있는 폰트와 그래픽 포맷에 제약이 있다. 이것을 요약한 것이 표 4.3이다. 물론 pk 비트맵 폰트로 변환하면 (pk 생성 유틸리티가 있거나 하다면) 모든 형식의 폰트를 사용할 수 있다 할 것이나, 여기에서는 고품위 pdf를 위한 윤곽선 글꼴의 사용만을 문제삼기로 한다.

ko \TeX /utf의 기본 글꼴은 은 글꼴 type 1 이므로, 폰트 이용상의 제약이 전혀 없다. 트루타입을 사용하려 하는 경우라면 표 4.3이 참고가 될 것이다.

7.3 pdf 한글 텍스트의 검색과 추출

pdf 파일을 제작하려 할 때 텍스트의 검색과 추출은 매우 중요한 문제 가운데 하나이다. 특히 개방적인 공개 문서의 작성이나 배포, 검색 엔진에서의 검색 가능성 등을 위해 한글 텍스트가 검색 가능하여야 한다. KTUGFaq/검색과 추출 페이지를 참고하라.

DVIPDFMx를 이용하면 이 문제를 쉽게 해결할 수 있다. ko \TeX 으로 작성한 pdf 문서는 DVIPDFMx로 pdf 변환한 경우 특별히 텍스트 검색과 추출을 제한하지 않으면 기본적으로 검색·추출이 이루어진다.

그런데, pdf \TeX 은 사실상 표준일 뿐 아니라 pdf \TeX 이 아니면 안 되는 경우도 더러 존재한다. pdf \TeX 최근 버전은 포스트스크립트 type 1 폰트의 경우 유니코드 한글 텍스트의 검색과 추출이 가능하다. 이미 말한 대로, 다음과 같은 코드가 있으면 된다.

```
\ifpdf
\input glyphtounicode\pdfgentounicode=1
\fi
```

그러나 트루타입 폰트에 대해서는 약간의 문제점을 가진다.

부가 패키지 dhucs-cmap 패키지는 이 문제를 해결하기 위한 것이다. 따라서 pdf \TeX 을 이용하고 트루타입 폰트인 경우 preamble에 다음과 같이 적어준다(여기서는 은 글꼴 트루타입을 예로 들겠다).

```
\usepackage{dhucs-cmap}
\pdfmapfile{=unttf-pdftex-kotex.map}
```

위의 문장은 반드시 $\text{ko}\TeX/\text{utf}$ 를 로드한 이후에 선언해야 하고 그렇지 않으면 효력이 없다. 두번째 행은 은 글꼴을 $\text{pdf}\TeX$ 에게 트루타입 형식으로 이용하도록 알려주는 것이다. 이 행을 생략하면 pk 비트맵 폰트를 만들어 이용하게 되므로 바라는 결과를 얻지 못할 수 있다. 만약 본문에서 사용하고자 하는 글꼴이 $\text{ko}\TeX$ 추가 폰트로 배포되는 은 글꼴 트루타입이 아니라 별도로 제작한 것이라면 자신의 폰트 설정을 적절한 방식으로 선언해 주어야 한다. `\pdfmapfile`의 인자로 주어지는 `map` 파일의 이름 앞에 붙은 `=`는 이미 해당 엔트리가 다른 `map`에서 스캔되어 있을 때, 지정된 `map` 파일의 엔트리로 교체하라는 의미이다. `+`를 지정하면 기존의 `map` 파일에 만약 해당 항목이 스캔되어 있다면 기존의 것이 사용되고 경고 메시지를 낸다. 이에 관한 사항은 $\text{pdf}\TeX$ 문서를 참조하라.

7.4 한글 책갈피

pdf 문서 제작에 있어서 한글 책갈피(bookmark)가 필요하다면, 다음과 같이 한다.

```
\usepackage[unicode,<driver>,...]{hyperref}
```

`hyperref` 패키지의 개선이 이루어짐에 따라, 한글 책갈피를 만들기 위해 별도의 복잡한 조치를 취할 필요가 없게 되었다. 다만, `<driver>`를 명시해주어야 한다는 데 주의하면 된다. `pdftex`, `dvipdfm`, `dvips` 등을 지정한다.¹⁶

제 8 절 찾아보기와 참고문헌

색인(찾아보기, index)을 만들기 위해 제공하는 유틸리티 `komkindex.pl`의 사용법에 대해서는 111 페이지 2 절을 보라. 이 유틸리티와 더불어 한글 색인 스타일인 `kotex.ist`를 제공한다.¹⁷

문헌목록 인용은 가능하다. $\text{Bib}\TeX$ 을 이용하여 문헌목록 데이터베이스에서 문헌 목록을 추출할 수도 있다.¹⁸

¹⁶혹 예전 `hangul-ucs`에서 제작한 문서에서 `dhucs-ucshyper` 패키지를 쓰는 경우가 있다면 이것을 `hyperref` 패키지를 이용하는 방식으로 소스를 수정하여야 한다. 이 패키지는 더이상 존재하지 않는다.

¹⁷ $\text{ko}\TeX/\text{euc}$ 에는 `hind.ist`가 제공된다. 이 둘은 호환가능하지 않으므로 $\text{ko}\TeX/\text{utf}$ 문서라면 반드시 `kotex.ist`를 색인 스타일로 지정해야 한다.

¹⁸다만, 한글화된 문헌목록 스타일이 충분하지 않아서, 최종적으로 약간의 수작업이 필요한 경우도 없지는 않다고 생각한다. 특히 저자-연도 방식의 인용에서 그러하다. 이 분야는 더 많은 연구와 개발이 필요한 지점이다.

참고문헌 인용은 너무나 많은 유형이 있어서 어떤 문제가 발생할 수 있는지를 완전히 시험해보지 못하였다. 그러나 `natbib` 패키지, `cite` 패키지와 `apacite` 패키지를 위한 코드를 내장하고 있어, 참고문헌 인용에서 발생하는 문제에 대한 지원을 위해 노력하고 있다.¹⁹

제 9 절 옛한글 구현

옛한글 문서의 조판은 한글 \TeX 의 꿈이었고, KTUG에서 2002년 이후 `dhhangul`과 같은 `Lambda` 패키지를 통하여 성공적인 실험을 행한 바가 있다.

현재 이른바 “한양PUA” 코드, 즉 유니코드 사용자 영역의 코드를 이용하여 옛한글 문자를 식자하는 것은, 이 영역에 완성형 옛한글 문자가 들어 있는 폰트를 지정하고 해당 코드를 입력하는 것으로 `dhucs`가 이미 잘 처리한다.²⁰ 그러나 이것은 엄밀한 의미에서 옛한글 문헌의 조판이라고는 말할 수 없다.

`dhucs-midkor` 패키지는 옛한글, 즉 1933년 이전 표기법 문헌을 식자할 수 있게 한다. 그러나 배포판의 매크로만으로 충분하지 않으며, 다음과 같은 준비가 더 필요하다.

1. 옛한글 폰트(`obatang.ttf` 또는 `ogulim.ttf`)
2. 옛한글 코드 변환 유틸리티
3. 옛한글 코드를 다룰 수 있는 편집기와 입력기

옛한글 문헌의 조판을 위해서 갖추어야 할 조건이 제법 까다롭고²¹ 이를 처리하는 과정이 편리하다고 할 수 없다²²고는 하나, 옛한글 문헌의 식자를 구현하고 있다는 자체가 매우 중요한 사실이다. 옛한글 조판에 관한 전반적인 사항, 이론적 배경과 구체적인 지침은 (이기황, 2007)을 참고하라.²³

¹⁹한 예로, `apacite` 패키지 충돌 문제의 보고와 그 해결에 관한 KTUG 게시판 4984 이하의 글타래를 참조하라. 이런 과정을 거쳐서 현재 `apacite` 패키지 등의 패키지에 대한 지원이 이루어지게 되었다.

²⁰이 매뉴얼에서는 언급하고 있지 않으나, 현재의 `hangul-ucs`는 이미 BMP 영역을 넘어서는 UCS4 문자를 식자할 준비를 갖추고 있다. 적절한 폰트와 입력코드만 있다면 유니코드 전 영역 문자의 식자가 가능하다.

²¹특히 옛한글을 처리할 수 있는 공개 폰트가 전혀 없다는 점은 매우 치명적이다.

²²코드 변환과 같은 약간의 전처리가 필요하다.

²³KC2007의 예제 문서에 옛한글 처리에 관련된 샘플이 포함되어 있다.

제 10 절 다른 패키지 및 클래스와의 호환성

10.1 \mathcal{AMS} - \LaTeX 클래스

`amsmath` 패키지를 \LaTeX 표준 클래스와 함께 쓰는 경우에는 아무 문제가 없지만 `amsart` 클래스, `amsbook` 클래스와 같은 \mathcal{AMS} - \LaTeX 클래스로 문서를 만들 때 한글을 사용하고자 한다면 부딪치는 문제가 몇 가지 있었다. `amsart` 클래스의 경우는 다음과 같이 한다.

```
\renewcommand\uppercasenonmath[1]{}
\usepackage{dhucs}
```

`amsbook` 클래스는 `\uppercase`가 아스키 코드에 대해서만 동작하기 때문에 발생하는 문제가 있는데, 이를 위하여 다음과 같은 코드를 필요로 한다.

```
\renewcommand\uppercasenonmath[1]{}
\makeatletter
\def\@makechapterhead#1{\global\topskip 7.5pc\relax
\begingroup
\fontsize{\@xivpt}{18}\bfseries\centering
\ifnum\c@secnumdepth>\m@ne
\leavevmode \hskip-\leftskip
\rlap{\vbox to\z@{\vss
\centerline{\normalsize\mdseries
\MakeUppercase{\chaptername}\enspace\thechapter}
\vskip 3pc}}\hskip\leftskip\fi
#1\par \endgroup
\skip@34\p@ \advance\skip@-\normalbaselineskip
\vskip\skip@ }
\makeatother
```

이 문제는 `amsbook` 클래스가 영어 이외의 언어를 표시하는 문자에 대해 배려하지 않아서 발생하는 것이므로, 위와 같이 수정하여 쓰는 것이 정당하다 할 것이다.

10.2 beamer 지원

`beamer` 패키지를 이용하여 PDF 발표 문서를 만들 때는, 다음과 같이 한다. 또는 간단히 `[unicode]` 옵션만 지정해도 될 것이다. `ko \LaTeX` 에서는 `[nojosa]` 옵션을 주어야 할 필요가 사라졌고, 따라서 `beamer` 패키지 등에서도 자동조사가 잘 작동한다.

```
\documentclass[hyperref={pdftex,unicode}]{beamer}
\usepackage{dhucs}
```

beamer 패키지 문서는 원칙적으로 pdf_{La}TeX으로 컴파일하여야 한다.

10.3 prosper/powerdot 지원

종래 prosper 패키지는 pdf 북마크를 hyperref 패키지를 통해서 만들지 않았기 때문에 한글 북마크의 구현이 쉽지 않았다. hyperref 패키지를 이용하는 최근의 prosper 패키지 또는 newprosper 패키지라면 다음과 같이 하여 북마크를 넣을 수 있다.

1. prosper.cls가 hyperref 패키지를 로드할 때 [unicode] 옵션이 활성화되도록 설정한다. 이 옵션은 hyperref에서 가장 먼저 주어지는 것이어야 하는데, 시스템의 hyperref.cfg에 \hypersetup{unicode=true}를 선언해두는 것이 가장 손쉬운 방법이다.
2. 글꼴은 type 1 글꼴을 이용한다. ko_{TeX}은 이미 type 1 글꼴이 기본 글꼴로 되어 있으므로 이 문제는 걱정할 필요 없다.

preamble은 다음과 같은 형식이 될 것이다.

```
\documentclass[ps2pdf.....]{prosper}
\usepackage{dhucs}
\SetHangulFonts{utbt}{utgt}{uttz}
\usepackage[dvips,unicode]{hyperref}
```

prosper에서 발전한 powerdot 패키지는 한글 사용에 문제가 없다. 다만 이 두 패키지는 모두 dvips, ps2pdf 방식으로 pdf 파일을 제작한다는 점만 염두에 두면 된다.

제 11 절 일본어와 중국어 문단

ko_{TeX}/utf 2.6 이후 버전은 dhucs-trivcj 패키지를 포함하였다. 이것은 한글 문서에서 약간의 일본어 또는 중국어 인용문을 처리하게 하려는 목적으로 제작된 것으로서 본격적인 다국어 문서를 처리하는 데는 아직 미흡하지만 크지 않은 규모의 일본어 중국어 문단은 나쁘지 않은 정도의 결과를 얻을 수 있을 것으로 생각하고 있다. 일본어 한자의 독음을 작은 가나 문자로 붙이는 것(루비)은 CJK 패키지의 일부인 ruby 패키지를 불러서 할 수 있다. 일본어 또는 중국어 문단을 식자하려면 dhucs-trivcj 패키지를 로드하여야 한다.

TeX というのは、市販ソフト顔負けのきれいな文書が作れる「文書整形ソフト」で、「ラテック」もしくは「ラテフ」など変った読み方をします。ちょっとカタイ言葉を使うと「組版(くみはん)ソフト」とも言います。組版とは、活字を組んでページに印刷するための版を作る作業のことで英語でタイプセッティング(type setting)と言います。

일본어와 중국어 글꼴은 TeXLive에 기본으로 포함되어 있는 wadalab-unicode 글꼴과 arphic 글꼴(type 1)을 이용한다. TeXLive를 full로 설치하면 당연히 시스템에 설치되어 있을 것이므로 별도의 글꼴 설치 절차 없이 바로 이용할 수 있다.²⁴ 다만 이 글꼴에는 굵은 글꼴이나 고딕체 글꼴이 별도로 갖추어져 있지 않다.²⁵ 일본 문자와 로마자가 이어질 때 필요한 ‘짧은 간격’은 tilde(~)로 표시할 수 있다. 이것은 CJK 패키지의 방식과 동일하다.

dhucs-trivcj가 제공하는 환경은 japanese와 chinese이다. 다만 chinese 환경은 간자체 환경과 번자체 환경을 각각 Schinese와 Tchinese로 구분할 수 있다. 기본값은 chinese가 간자체 환경으로 되어 있지만 만약 번자체 환경을 chinese로 쓰고 싶다면

```
\let\chinese\Tchinese
\let\endchinese\endTchinese
```

라고 선언하면 된다. 다른 문서와의 호환을 위해서 번자체 환경을 그대로 Tchinese라고 쓰는 것도 좋은 방법이다.

다음 표는 dhucs-trivcj에서 사용하는 각 언어별 폰트의 이름이다.

	trivcj		trivcj-adobe	
	mj	gt	mj	gt
일본어	min	-	jpmj	jpgt
중국어간자	gbsn	-	cnmj	cngt
중국어번자	bsmi	-	ctmt	-

간단한 중국어 문장 하나를 식자해보자.²⁶

²⁴KC2006과 같은 시스템에서는 이 글꼴을 설치할 수 있는 설치 패키지를 별도로 제공할 것이다.

²⁵Hangul-ucs에서 쓰이던 Adobe Reader의 opentype을 위한 설정을 사용하기 원한다면, dhucs-trivcj-adobe라는 꾸러미가 별도로 준비되어 있으므로 그것을 이용하라. 다만 이 때는 \usepackage{dhucs-trivcj-adobe}라고 선언해야 하며, pdf 제작에 DVIPDFMx 만을 사용할 수밖에 없다는 점을 기억해두자.

²⁶<http://wikka.ctex.org/TeX> 웹사이트에서 인용함.

[高质量的输出] TeX 遵循传统的排版规则，以排版的质量为最重要的目标。如果你把 TeX 的输出结果和用其它的排版软件排版相同的文本所得到的结果加以比较，你就会发现其中的区别。

[超常的稳定性] 自从 TeX 出现以来，只有一些微小的改动。也就是说，十几年前的 TeX 文件用现在的 TeX 系统排版得到的结果与十几年前得到的结果是一样的。稳定性还体现在 TeX 系统极少会崩溃，可以处理任意大小的文件，即使你的计算机的内存很少，TeX 也可自如的工作。

ko.TeX의 dhucs-trivcj 는 pdfTeX, DVIPDFMx, dvips 어떤 드라이버로도 잘 처리할 수 있다.

제 5 장 memhangul과 oblivoir

제 1 절 소개

memoir 클래스는 Peter Wilson 씨가 작성한 것으로 큰 규모의 문서를 조판하는 클래스이다. \LaTeX 표준 클래스 중에서는 book이 있는데, 이 book 클래스를 실제 책의 작성에 사용하기 위해 손보아야 할 부분이 상당히 많다. memoir는 book과 비슷한 착상에서 출발했으나, 여기저기서 제안된 book을 수정하는 기능을 하나의 클래스로 합쳐둔 것에 가깝다.

memhangul-ucs 패키지는 dhucs의 한글 식자 능력을 memoir 클래스에서 이용하기 위해 작성된 것이다. memhangul 자체가 dhucs를 부르며, 그 위에 memoir에 맞는 여러 정의들을 추가하고 있다. memhangul은 원래 독자적인 스타일로 제작되었으나, 2007년 이후 hangul-ucs에 합쳐져 ko \TeX 으로 계승되었다. 이러한 배경을 가지고 있는 까닭에 oblivoir는 오직 유니코드 한글만을 처리할 수 있으며, EUC-KR 한글 코드로 문서를 작성할 수 없다.

oblivoir는¹ memoir 클래스를 이용하면서도 마치 article 클래스의 느낌으로 편리하게 한글 문서를 작성하도록 한 클래스이다. oblivoir는 한글 문서 작성을 목적으로 만들어진 것으로서, 한글화 설정 등을 별도로 할 필요가 없게 되어 있다. 이 클래스의 의의를 우리는 “표준적인 한글 문서작성 서식”을 제공하는 데 두었다. 즉, 대부분의 사용자 설정의 기본값을 미리 제공하지는 것이다. 만약 이 기본값에 만족하지 못한다면 memoir 문법으로 손쉽게 사용자화할 수 있게 하고 있다.

dhucs, memoir, memhangul, oblivoir가 상호 의존하는 관계는 그림 5.1에 예시한 바와 같다.

제 2 절 memhangul

memoir 클래스 자체가 방대한 기능으로 이루어져 있으며 memhangul 역시 상당한 추가 옵션과 매크로를 가지고 있다. 이 기능을 모두 설명하는 것은 별도의 매뉴얼을

¹이 이름은 oblivescence(망각)라는 단어와 memoir를 합쳐서 만든 신조어이다. 원래 memoir 클래스의 memoir는 논문, 책 등을 의미하는 단어였지만 여기에서는 ‘기억’이라는 의미와의 연관성을 고려하여 말장난을 한 것이다.

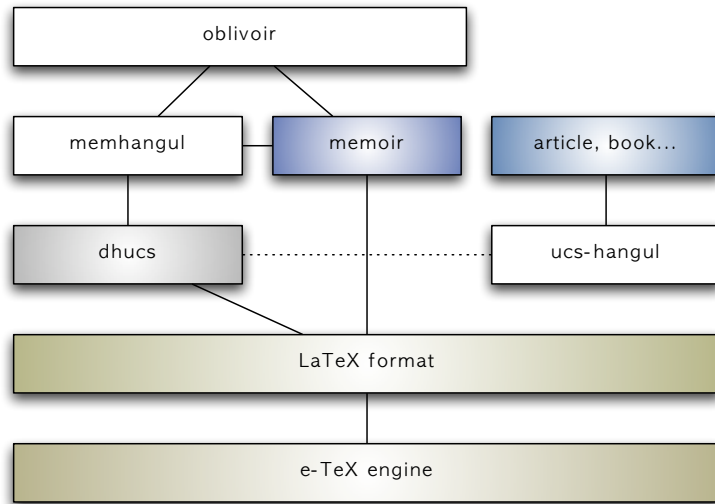


그림 5.1: oblivoir tree

참고하는 것이 좋다. KTUGFaQ/MemhangulClass 페이지를 보라. memucs-manual 이라는 이름의 한글판 memoir 매뉴얼은 memoir와 memhangul을 이용하려 할 때 필독서이다. 특히 memhangul 패키지에는 한글 타이포그래피를 충족시키기 위한 많은 추가 기능들이 들어 있으며, 이것은 옵션으로 또는 선언으로 조절할 수 있도록 설계되어 있다.

여기서는 간단한 preamble 하나를 예시하는 것으로 만족하려 한다.

```

\documentclass{memoir}
\usepackage{memhangul-ucs}

```

제 3 절 oblivoir

oblivoir 역시 다양한 옵션과 복잡한 기능을 구현한 설정 패키지들로 이루어져 있다. oblivoir의 사용법 등에 대해서는 KTUGFaQ/Karnes/Oblivoir/FAQ 페이지를 참고하라.

3.1 간단한 문서 만들기

```

\documentclass{oblivoir}

```

위의 클래스 선언으로 한글 문서를 즉시 작성할 수 있다. pdf_TTeX과 DVIPDFMx 드라이버는 pdf_latex으로 컴파일하느냐 latex으로 하느냐에 따라 결정되며, dvips 드라이버를 사용하려 한다면 [dvips] 옵션을 명시해주면 된다.

oblivoir는 한글 문서 작성시 pdf bookmark를 만드는 것이 default이다. 만약 pdf bookmark를 만들지 않으려 한다면 [nobookmarks] 옵션을 명시해준다.

[10.5pt]라는 실험적인 옵션도 있는데, 이것은 본문의 기본 글꼴 크기를 10.5 포인트로 맞추어주는 것이다. 10포인트보다는 크고 11포인트보다는 작은 기본 글꼴 크기를 원하는 경우 사용할 수 있다. 또, \tt, \sf 같은 \LaTeX 2.09 글꼴 명령을 사용하려면 memoir 옵션인 [oldfontcommands]를 지시한다.

3.2 용지, 여백, trim mark

memoir의 문서 옵션인 종이 크기 옵션 [a4paper] 등을 지시하는 것이 가능하다. 이에 더하여, 용지와 여백을 일괄 조정할 수 있는 fapapersize 패키지가 제공되는데, 이것은 표준 \LaTeX 클래스와는 함께 쓸 수 없고, memoir와 oblivoir에서만 유효하다.

fapapersize 옵션으로 [db14x6] 또는 [newmum] 등을 지정할 수 있으며, 사용자 용지여백 설정 명령인 \usefapapersize 명령을 사용할 수 있다.

```
\usepackage{fapapersize}
\usefapapersize{*,*,40mm,*,40mm,*}
```

stocksize를 papersize보다 크게 하면 trim mark를 그릴 수 있다. 이것은 memoir의 trim mark를 이용하는 것이므로 memoir 매뉴얼을 참고하면 쉽게 이해할 수 있다. oblivoir 옵션 [showtrims]를 선택하면 된다.

3.3 글꼴 선택

본문 글꼴을 선택할 수 있는 명령 \SelectHfonts가 제공된다. 사용법은 다음과 같다. 인자는 두 개이며, 첫번째 것은 한글, 두번째 것은 한자 글꼴을 각각 rm/sf/tt에 대하여 쉼표(,)로 구분한다.

```
\SelectHfonts{<rm>,<sf>,<tt>}{<rm>,<sf>,<tt>}
```

*를 이용하여 다음과 같이 간략하게 쓸 수 있다.

```
\SelectHfonts{kxibt,kxgt,*}{*}
```

3.4 기타

oblivoir는 자체 명령뿐 아니라 memhangul의 명령, memoir의 명령, \LaTeX 의 명령, \TeX 의 명령을 모두 사용할 수 있다. 위에 소개한 것 말고도 많은 추가적인 기능이 있으므로 매뉴얼을 참고하라.

제 III 편

ko.T_EX/euc

제 6 장 ko.TeX/euc 의 의의와 사용법

제 1 절 ko.TeX/euc 의 의의

1.1 문자의 범위

ko.TeX/euc는 EUC-KR 한글만을 처리한다. EUC-KR 한글이라고 통칭하는 이 한글 인코딩 방식은 특히 KS X 1001의 완성형 한글과 한자 및 기호문자를 지칭하는 것으로 사용하겠다. Windows 운영체제의 기본 한글 인코딩은 CP949라고 불리는 것으로 이 EUC-KR 완성형 한글을 확장한 “확장완성형” 인코딩이다.

EUC-KR은 한글 2350자만을 정의하고 있으므로 유니코드 현대 한글 음절문자 11172자의 일부밖에는 표현하지 못한다. 우리의 언어 문자생활에 필요한 거의 대부분의 문자가 들어 있다고 하나 한글의 특성 중 하나인 자유로운 자모 결합에 의한 표현력을 희생하지 않을 수 없으며, (비록 표준 정서법에는 어긋나더라도) 특별한 목적을 위해서 표현하고자 하는 일부 글자를 찍을 수 없다는 한계를 갖는다.

이러함에도 불구하고, EUC-KR의 필요성도 많다. 가장 많은 사람이 사용하는 윈도우즈 운영체제에서 아무런 불편없이 입출력할 수 있다는 점, 그리고 무엇보다 기존에 이 완성형 한글로 작성된 문서와 문헌이 워낙 방대하다는 점이다.

ko.TeX이 EUC-KR 지원을 하는 이유는 궁극적으로 이 때문이다. 기존에 작성된 문서들에 대한 호환성을 유지하는 것. 또한, 장치 ko.TeX은 EUC-KR 문서를 안전하게 유니코드로 포팅하는 방법을 발전시켜¹ 장기적으로 ko.TeX/utf로의 이행을 권장하기 위해서이다.

완성형 한글의 범위는 부록을 참고하라.

1.2 ko.TeX/euc의 한계

ko.TeX/utf와는 달리 ko.TeX/euc가 호환을 위해서 제공되기 때문에 ko.TeX/utf의 많은 기능들을 생략하거나 지원하지 않는다. 그 때문에 이 두 형식의 문서가 출력 결과에 있어서 동일하지 않다.

ko.TeX/utf의 몇 가지 옵션, 예를 들면 `nonfrench`, `finemath` 등은 지원하지 않는다. ko.TeX의 자랑 중 하나인 미세 간격 기능은 euc 버전에서는 동작하지 않는다.

¹아직까지는 이 호환이 완전히 제공되지 않는다.

1.3 하위 호환성

종래의 한글 문서들이 `hangul` 또는 `hfont` 패키지를 이용하고 있음이 엄연하고, 따라서 이 패키지들에 대하여 하위호환성을 제공한다. 그러나 되도록 이 문서에서 설명하는 바와 같이 `kotex` 기본 패키지를 이용하는 방식으로 이행해가는 것이 바람직하지 않겠는가 기대한다. `\usepackage{hfont}`는 `\usepackage[euc]{kotex}`으로, `\usepackage{hangul}`은 `\usepackage[euc,hangul]{kotex}`으로 바꾸어 쓰면 거의 동일하게 동작할 것이다.

한글 라텍의 특징 중 하나였던 글꼴 선택 명령 체계는 여전히 지원된다. 그러나 유니코드와의 호환성을 위하여 이 부분에 대한 설명은 최소한에 그치고 호환성 높게 문서를 작성하는 방법을 위주로 설명하고자 한다.

기본 글꼴은 더이상 UHC 폰트를 사용하지 않게 되었다. 그러나 사용자 수준에서 글꼴 처리에 대해서는 걱정하지 않아도 좋다.

종래 `hangul` 패키지와 함께 제공되던 `khyper` 패키지는 제외되었다. `ko.TeX/euc`가 `hyperref`을 잘 지원하며, 심지어 필수적으로 요구하기까지 하는 까닭이다. 이 부분이 아마 가장 많이 달라진 부분 중 하나일 것이다.

1.4 사용설명서에 관하여

이 사용설명서의 `ko.TeX/euc` 부분은 한글 라텍의 “한글 라텍 길잡이” 문서를 거의 그대로 인용한 것이다.

제 2 절 ko.TeX/euc의 사용 설명

이 단원에서는 라텍 자체의 기본적인 사용법에 대한 설명을 피하고, 한글 라텍에 관계되는 부분만을 간단히 다루려 한다.

2.1 한글 문서의 틀

문서는

```
\documentclass[추 가선택]{문서종류}
```

로 시작한다. 추 가선택에는 라텍의 문서종류(class)에 종속적인 추가 선택을 지정한다.

문서 종류 지정 후에 한글 라텍 꾸러미 사용을 지정한다.

```
\usepackage[euc,추 가선택]{kotex}
```

한글 라텍의 추 가선택의 종류는 다음과 같다.

- `euc`: EUC-KR 한글을 처리하는 `ko.TeX/euc`에서는 이 옵션이 가장 처음에 있어야 한다.
- `hanja`: 문서의 단원명이나 낱자가 한자로 표기된다.
- `hardbold`: METAFONT에 의해 만들어지는 두꺼운 글씨체를 쓰도록 한다.
- `nojosa`: 자동 조사 처리 기능을 억제하여 텍의 기억 용량 사용을 감소시킨다.

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

- `\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[euc]{kotex,a4}
```

에서는 지정된 추가 선택 `euc`를 두 꾸러미 `kotex`과 `a4`가 모두 처리할 수 있어야 한다. `a4` 꾸러미는 `euc`이라는 추가 선택을 처리할 수 없으므로 오류가 발생하게 된다. 다음과 같이 각기 따로 지정하거나

```
\usepackage[euc]{kotex}
```

```
\usepackage{a4}
```

아니면 문제가 되는 추가 선택 `euc`를 `\documentclass`에서 전역적 추가 선택으로 지정하면 오류가 발생하지 않는다.

```
\begin{document}
```

로 문서를 시작하여 쓰고자 하는 글을 작성한 뒤

```
\end{document}
```

로 문서 작성을 마친다. (그림 6.1을 참조)

이제 `latex`이나 `pdflatex`으로 파일을 돌리면 된다.

2.2 문서의 모양새 설정

전반적인 문서의 종류를 결정하기 위해 다음 문장을 단 한 번 맨 처음에 쓴다.

```
\documentclass[추가선택]{문서종류}
```

<code>\documentclass[추 가선택]{문서 종류}</code>	
<code>\usepackage[euc,추 가선택]{kotex}</code>	← 한글 문서 특잡기를 위해 [hangul] 추가선택을 추가할 수 있다.
<code>\usepackage[추 가선택]{꾸러미}</code>	← 필요에 따라 사용될 각종 꾸러미를 지정한다.
<code>...</code>	← 꾸러미뿐만 아니라 문자를 축약하지 않는 다른 명령들은 여기에서 선언할 수 있다.
<code>\begin{document}</code>	← 이전까지를 전문 영역(preamble)이라고 한다.
나타의 말이 중국과 달라서 한자로 는 서로 통하지 아니하므로 이런 가 답으로 어리석은 백성들이 말하고 자 하는 바가 있어도 마침내 제뜻 을 능히 펴지 못하는 사람이 많으 니라. 내가 이를 불쌍히 여겨 새로 스물여덟자를 만드나니	← 문서의 작성은 \TeX 의 관습에 따른다. 즉, 중첩된 공간 문자는 하나로 처리된다. 줄바꿈 문자가 한번 나오면 공간 문자로 처리되거나, 두 번 이상 중첩되면 하나의 줄바꿈 문자 역할을 한다. 등등.
<code>\end{document}</code>	← 이 명령이 나온 후의 모든 문장은 무시된다.

그림 6.1: 한글 라텍 문서의 기본 구조

문서종류(class)에는 article 이외에 report, book, letter, slides 등이 있다. 각 문서종류는 추가선택에 의해 모양새가 달라질 수 있다. 즉, 종이의 규격을 설정하는 a4paper, letterpaper, ... 및 글자의 크기를 정해주는 10pt, 11pt, 12pt 등등이 있다. 일반 영문 라텍의 문서 종류 설정은 (Kopka, 1991; Lamport, 1985; Mittelbach et al., 2004)에 잘 설명되어 있다.

위와 같이 라텍에서 제공하는 기본적인 문서 종류 이외에 추가로 꾸러미(package)를 사용하고자 할 경우에는 `\usepackage` 명령을 쓴다.

`\usepackage[추 가선택]{꾸러미}`

우리말의 처리와 같이, 라텍에서 제공되지 않는 기능을 쓸 경우에 kotex.sty을 쓰는 것과 같다. 각 꾸러미는 자체의 추가선택이 있다. 각종 꾸러미의 사용 방법 및 추가선택의 종류는 (Mittelbach et al., 2004)에 자세히 설명되어 있다.

한글 라텍으로 우리말을 처리하게 하려면 다음과 같이 한다.

```
\usepackage[euc]{kotex}
```

`kotex.sty`를 `euc` 이외의 추가선택 없이 쓰면 (종래 `hfont.sty`를 쓴 것과 같이) 문서 틀잡기가 우리말에 어울리는 한글화를 전혀 반영하지 않고 단순히 한글 및 한자, 상징 기호를 포함한 문서에서 우리말 처리가 제대로 될 수 있게만 한다. 자동조사는 동작하지만, 자동조사의 동작을 위해서는 `hyperref` 패키지를 필요로 한다. 만약 순수하게 한글을 식자할 의도로 자동조사를 필요로 하지 않는다면 `nojosa` 옵션을 주면 된다.²

영문 문서와 달리 우리말 문서에서는 줄간격을 특별히 취급해야 한다. 이는 우리말 글자들이 영문자와 비교해서 모양이 복잡함으로 인해 발생하는 문제로서, 라텍스의 애초값으로 우리말 문서를 만들 경우 각각의 페이지가 글자로 밀집된 인상을 받는다. 이는 글을 읽는 독자의 시각을 분산시키는 악영향을 미치므로 다음과 같은 명령으로 줄간격을 조절하는 것이 좋다.

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

또는

```
\linespread{1.3}
```

이 명령은 줄간격을 지정하는 레지스터인 `\baselineskip`을 1.3배로 증가시키는 작용을 하며 `\normalsize`와 같이 글자체의 크기 바꿈 명령이 나타날 때 이러한 변경이 효력을 발휘한다. `koTeX/euc`에 `[hangul]` 추가선택을 더하여 한글 문서 틀잡기를 활성화하면 `\baselinestretch`의 기본값이 1.333으로 자동 설정된다. 이 설정은 문서의 어느 곳에서든지 사용자의 취향에 따라 변경될 수 있다. 줄간격을 좀더 쉽게 제어할 수 있도록 하는 `hsetspace` 패키지가 제공된다. 이것은 `setspace` 패키지를 불러들여서 문서 내의 각주나 떠다니는 개체 내의 줄간격을 재설정하는 등 몇 가지 추가적인 기능을 제공한다. 98 페이지 2.3 절을 참고하라.

2.3 글자체 선택

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

이러한 글자체 바꿈을 위해 라텍스는 여러가지 글자체 바꿈 명령을 제공한다. 라텍스 209에서 제공했던 `\rm`, `\sf`, `\sl`, `\it` 등등의 글자체 바꿈 명령들은 모두가 같은

²그러므로 엄밀히 말하면 예전의 `hfont`와 동일한 것은 `[nojosa]{kotex}`이다.

차원에서 글자체를 바꾸기 때문에 여러가지 단점을 가지고 있었다. 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 아니라 `\rm`의 `\sl`이 된다. 이는 이런 글자체 바꿈 명령이 일단 글자체를 애초값으로 복귀시킨 후 글자체를 바꾸기 때문이다. 이러한 단점을 보완하기 위해 1989년부터 Frank Mittelbach와 Rainer Schöpf는 “새로운 글자체 선택 방식”(NFSS: New Font Selection Scheme)을 고안하기 시작하여 현재 라텍 버전 2ε에 채택되어 있는 NFSS를 만들었다.

2.3.1 영문 글자체 선택 방법

NFSS는 글자체를 다섯가지 요소로 구분하여 이 요소들을 각각 독립적으로 선택할 수 있게 한다. 그 다섯가지 요소는 다음과 같다.

글자체 부호화(encoding) 다른 알파벳을 쓸 때 각 나라별로 `codepage`가 정해져 있다. 라텍에서는 다른 프로그램과 마찬가지로 각 글자와 0-255 사이의 글자체 위치를 연관시킴으로써 이를 해결한다. 라텍에서는 이를 “부호화 방법”이라 한다.

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

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

글자체 무게(weight) 및 폭(width) 무게와 폭은 서로의 조합으로 하나의 글자체 속성을 형성한다. 무게는 글자의 두께로서 결정되며 “보통 두께(medium)”, “가벼운 두께(light)”, “무거운 두께(bold)”... 등으로 나뉜다. “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 세분된다. 폭은 “보통의 폭(medium width)”, “확장된 폭(extended width)” 등이 있다.

글자체 크기(size) 글자체 크기는 인쇄기의 점을 단위로 표현된다. 인쇄기의 점을 텍에서는 pt라는 단위로 나타내며 1인치는 72.27pt이다. 라텍에서는 글자체 크

기가 1.2의 승으로 구별된다. 이렇게 정한 배경에는 후에 축소 복사를 할 때의 편의함이 있다. 즉 A5 크기의 책자를 만들 경우, A4 종이에 $1.44 = \sqrt{2}$ 의 글자체 크기를 선택하여 인쇄한 후 A5 크기로 축소한다.

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

<p>몇몇 단어들은 <code>\begin{bfseries}</code> 두 겹 겹 <code>\end{bfseries}</code> 인쇄된다.</p>	⇒	<p>몇몇 단어들은 두 겹 겹 인쇄된다.</p>
---	---	---

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

문서의 글자체 가족에는 로만(roman) 체, 산세리프(sans serif) 체, 타자(type-writer)체가 있다. 이들은 각각 다음의 명령형과

`\textrm \textsf \texttt`

선언형으로

`\rmfamily \sffamily \ttfamily`

선택된다. 로만 글자체는 일반적으로 문서의 기본 글자체로 선언되어 있다.

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

`\textmd \mdseries`

`\textbf \bfseries`

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

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

`\textup \upshape`

`\textit \itshape`

`\textsc \scshape`

`\textsl \slshape`

이 중에 굵은 모양과 곧은 모양 사이의 전환은 `\emph` 명령 및 `\em` 선언으로 이루어진다.

글자체 크기 선택을 위해서는 10 가지의 선언형이 사용된다. (명령형은 없다.)

<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 달라진다. 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 받는다.

라텍에서 애초값은 표 6.1 과 같이 설정되어 있다.

2.3.2 우리말 글자체 선택 방법의 개요

HFSS³는 NFSS를 바탕으로 두고 있다. 우리말 부호화의 특성에 따라 한국어 부호화(encoding) 방식을 지정하는 `\hfontencoding`과 한국어 글자체 가족(font family)을 지정하는 `\hfontfamily`가 추가 되었으며 글자체 선택 방식도 우리말 글자체에 맞도록 변경되었다.

글자체 부호화 우리말의 부호화는 H로 설정되어 있고 NFSS에서 사용되는 글자체 부호화 설정 모듬 명령 `\fontencoding`을 우리말화한 `\hfontencoding`을 쓴다. 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 한다.

³Hangul Font Selection Scheme

표 6.1: 라텍스에서 글자체 애초값	
내형 글자체	애초값
<code>\encodingdefault</code>	OT1
<code>\familydefault</code>	<code>\rmdefault</code>
<code>\seriesdefault</code>	m
<code>\shapedefault</code>	n
<code>\rmdefault</code>	cmr
<code>\sfdefault</code>	cmss
<code>\ttdefault</code>	cmtt
<code>\bfdefault</code>	bx
<code>\mddefault</code>	m
<code>\itdefault</code>	it
<code>\sldefault</code>	sl
<code>\scdefault</code>	sc
<code>\updefault</code>	n

글자체 가족 NFSS에서와 같이 우리말의 글자체 가족은 글자체의 설계 원칙에 따라 구분이 되고, 우리말의 특성에 의해 두 요소로 표현된다.

첫번째 요소는, `\ksfamily`로서 우리말의 상징 문자와 한글 및 한자를 구별한다. 이 모든 명령도 사용자의 입장에서는 쓸 일이 거의 없고, 주로 한글 라텍의 내부 처리 과정에서 사용된다.

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

글자체 무게 및 폭 작동 방식과 내용이 NFSS와 동일하다. 추가된 사항은 글자체 시리즈의 접두사 S로서 이는 우리말의 `\bfseries`가 `softbold`로 처리되도록 한다.

글자체 모양 NFSS와 같다.

글자체 크기 NFSS와 같다.

2.3.3 우리말 글자체의 가족 선택

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

```
\kscfamily{상징문자}{한글}{한자}
```

이는 각각

```
\symboldefault
```

```
\hanguldefault
```

```
\hanjadefault
```

로 설정되어 있으며 그 값은 각각 s, w, h이다.

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 한다. (왼쪽은 명령형, 오른쪽은 선언형)

위와 같은 모든 명령들은 우리말 가족을 바꾸어 주는 모든 명령 `\hfontfamily`를 사용한다.

```
\hfontfamily{우리말가족}
```

우리말 가족에는 다음과 같은 글자체가 있다.

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)				

위와 같은 우리말 가족들은 상징 문자/한글/한자의 세 가족을 대표하는 외형 가족으로서, 문서에서 우리말 가족을 tz으로 지정하였을 경우 한글 라텍은 한글/상징 문자/한자의 순으로 각각

wtt, wtt, wtt

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

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

tz의 경우 내형 가족과의 연관은 다음과 같은 명령으로 이루어진다.

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

2.3.4 우리말 글자체 시리즈

문서의 글자체 시리즈의 선택에는 NFSS가 그대로 적용되며 라텍을 쓸 때는 한글 라텍 버전 0.96까지 사용되었던 `softbold`를 위해 글자체 시리즈의 접두사 S가 추가되어 보통 두께의 글자체를(mdseries) 세 번 중첩 인쇄하는 기능을 한다. 그러므로 라텍 문서에서 다음의 명령은

`\renewcommand{\bfdefault}{Sbx}`

혹은

`\fontseries{Sbx}`

우리말의 글자체 두께를 `softbold`로 처리하도록 하며 영문 글자체의 두께는 bx가 되도록 한다. `kotex.sty`의 추가 선택 `softbold`는 위와 같은 기능을 애초값으로 설정한다.

2.3.5 우리말 글자체의 모양과 크기

우리말 글자체의 모양/크기는 NFSS와 같다.

2.3.6 우리말 글자체의 애초값

문서의 애초 글자체는 NFSS에서와 같이 다음의 명령 / 선언형으로 선택된다.

```
\textnormal \normalfont
```

한글 라텍스에서 애초값은 다음과 같이 설정되어 있다.

	한글 라텍
\hencodingdefault	H
\encodingdefault	
\symboldefault	s
\hanguldefault	w
\hanjadefault	h
\mjdefault	mj
\gtdefault	gt
\tzdefault	tt
\hfamilydefault	\mjdefault

위와 같은 애초값들은 운영 체제 관리자가 hfont.cfg에서 변경 설정할 수 있고, 사용자도 문서의 서두에서 재정의할 수 있다.

2.3.7 그외 우리말 글자체의 모듬 명령

한글 라텍 버전 0.96까지 써 오던 두 글자로 구성되는 글자체 바꿈 명령은 기본 글자체 이외에는 모두 삭제되었다. 이들은 라텍 버전 209 식의 글자체 바꿈 명령에 해당하며 사용을 회피하는 것이 좋다.

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

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

	한글 라텍
\명조	\textmj \mjfamily

\고딕	\textgt \gtfamily
\탁자	\texttz \tzfamily
\그레픽	\hfontfamily{gr}
\궁서	\hfontfamily{gs}
\신문	\hfontfamily{sh}
\필기	\hfontfamily{pg}
\펜글씨	\hfontfamily{pn}
\펜홀림	\hfontfamily{ph}
\봄글씨	\hfontfamily{bm}
\옛글	\hfontfamily{yt}
\자모명조	\hfontfamily{jmj}
\자모고딕	\hfontfamily{jgt}
\자모노벨	\hfontfamily{jnv}
\자모소탁	\hfontfamily{jsr}

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

\usefont는 NFSS의 모든 명령으로서 글자체의 부호화와 가족, 시리즈 및 모양을 한번에 정할 수 있도록 한다.

\usefont{부호화}{가족}{시리즈}{모양}

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

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

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

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

\DeclareErrorHFont{부호화}{가족}{두께}{모양}{크기}

\DeclareHFontSubstitution{부호화}{가족}{두께}{모양}{크기}

2.3.8 ko_{TeX}/utf와 ko_{TeX}/euc의 글자체 선택 방식

ko_{TeX}/utf는 HFSS를 채택하지 않고 있다. 따라서 \hfontfamily나 \hfontseries 등에 상응하는 명령이 없다. 그 대신 한글 폰트의 글꼴 가족을 지정하되, 다른 폰트 속성 등은 NFSS를 따르도록 하고 있다.

또한 글꼴 가족에 대한 개념도 달라서, euc 버전이 한글/심벌/한자를 묶어서 하나의 가족으로 부르는 반면, utf에서는 한글 글꼴 가족과 한자 및 기타 문자 글꼴 가족으로 분리시켜 두고 이들을 대응하는 방식을 선택하였다.

예를 들어, 본문의 rm/sf/tt 글꼴 세트를 바꾸기 위해서 euc 버전에서는

```
\MapHangulFamily{mj}{XXbt,XXbt,XXbt}
\MapHangulFamily{gt}{XXgt,XXgt,XXgt}
\MapHangulFamily{tz}{XXtz,XXtz,XXtz}
\mjfamily, \gtfamily, \tzfamily
```

와 같이 지정해야 할 것이나, 이것을 utf 버전에서는

```
\SetHangulFonts{XXbt}{XXgt}{XXtz}
\SetHanjaFonts{XXbt}{XXgt}{XXtz}
```

과 같이 할당한 다음, \rmfamily, \sffamily, \ttfamily로 쓰도록 하고 있다. euc 버전과 utf 버전 사이의 호환성에 가장 큰 걸림돌이 이 폰트 선택 명령이다. 한글 라텍 방식대로 한다면 듣도보도 못한 \XXfamily가 나오지 말라는 법이 없는데, 이것을 일일이 모두 지원할 수 없는 까닭이다.

실제로, HFSS로 인하여, \XXfamily 명령을 남발하게 되면 이것은 심각한 호환성의 문제를 불러일으킨다. 다른 곳에서는 컴파일이 불가능한 상황이 생기는 것이다. 우리는 rm, gt, tz라는 세 가지 한글 글꼴 패밀리로 너무 많다고 생각한다. 그러므로, 이 세 종류의 글꼴 가족 이외에는 반드시 \hfontfamily{XX}\selectfont를 일관되게 쓰는 것이 좋으며, 본문 글꼴을 교체하려 할 때는 \MapHangulFamily를 사용하도록 할 것을 권장한다. 또한 \textmj와 같은 명령은 아예 일체 사용하지 않는 것이 좋다고 생각한다. euc 버전에서도 \rmfamily, \sffamily 등 NFSS 명령만을 사용해서 문서를 작성하는 것이 좋다.

2.4 우리말 숫자 모듬 명령

한글 라텍에서는 우리말 숫자를 숫자 환경에서 쓸 수 있도록 우리말 숫자 모듬 명령을 제공한다. 우리말 숫자 모듬 명령에는 다음과 같은 것들이 있다.

```
\jaso(ㄱㄴㄷㄹ) \gana(가나달라)
\ojaso(㉑㉒㉓㉔) \ogana(㉑㉒㉓㉔)
\pjaso((ㄱ)(ㄴ)(ㄷ)(ㄹ)) \pgana((가)(나)(다)(라))
\onum(①②③④) \pnum((1)(2)(3)(4))
\oeng(@bC@) \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.

\begin{enumerate}	
\item 한글	가) 한글
\begin{enumerate}	
\item 현대어	ㄱ. 현대어
\item 고어	ㄴ. 고어
\end{enumerate}	
\item 漢字	나) 漢字
\begin{enumerate}	
\item 韓國語	ㄱ. 韓國語
\item 中國語	ㄴ. 中國語
\item 日本語	ㄷ. 日本語
\end{enumerate}	
\item 상징 기호	다) 상징 기호
\begin{itemize}	
\item ㉠	㉠ ㉠
\item ₩	₩ ₩
\item ㄹㅅㅁ	ㄹㅅㅁ ㄹㅅㅁ
\end{itemize}	
\end{enumerate}	

나열환경의 항목 머리에 대하여 ko_{TeX}/utf는 좀더 편리한 방식을 지원한다. 그렇지만 ko_{TeX}/euc에 대해서 별도의 추가 지원 패키지를 마련하지 않았다. 즉 ko_{TeX}/euc에서는 enumerate 패키지의 환경 옵션으로 한글 항목머리를 지시할 수 없다.⁴

utf

2.5 우리말 각주 판짜기

영문 라텍에서의 기본적인 각주 판짜기 방식을 사용하면 각주 번호가 표시되는 각주 첫 줄이 들어 쓰여진다. 각주 번호에는 수학적식에서 먹수를 나타내는 윗첨자가 사용

⁴enumitem 2.0 이후 버전은 항목 머리 옵션을 추가하는 것이 매우 쉬우므로 ko_{TeX}/euc에서도 설정해서 쓸 수 있을 것으로 생각한다.

되고 각주 번호와 각주문을 구분하는 공간이 없다. 이와 같은 판짜기 방식 이외에도 출판된 우리말 문서를 보면 다른 방식의 판짜기를 볼 수 있다.

(이종운, 1991)은 각주면을 본문의 왼쪽에 맞추는 방식과 각주면을 본문의 안쪽으로 들여 넣는 방식으로 각주면 판짜기를 대분하고 각주문의 첫줄을 들여쓰기와 내어쓰기에 의해 다시 세분하는 방식으로 각주 판짜기를 분류한다.

한글 라텍의 각주 판짜기는 영문에서의 방식을 그대로 사용한다. 그러나 사용자의 선호도에 따른 각주 판짜기 방식을 쉽게 적용할 수 있도록 `hangulfn.sty` 패키지를 따로 제공한다. 93 페이지를 보라.

2.6 조사 이형태, 매개 모음, 지정사 탈락

`\ref` 나 `\pageref` 등의 교차 참조와 `\cite` 등의 문헌 인용을 사용하면 숫자나 이름표가 자동으로 생성된다. 이 경우 문서를 작성하고 있는 순간에는 이 숫자나 문자가 무엇인 지를 알 수 없다. 그러므로 그 다음에 오게 되는 주격 조사가 ‘가’가 될 지 아니면 ‘이’가 될 지를 결정할 수 없다. 사용되는 변이형은 자동으로 생성되는 숫자의 음가나 이름표의 마지막 음절이 무슨 받침을 갖고 있는가에 따라 다르다. 매개 모음 “으”의 삽입이나 지정사 “이”의 탈락도 같은 경우다. `ko.TeX/euc`는 다음과 같은 음운 이형태를 자동 처리할 수 있도록 했다.

은/는	이/가	을/를	와/과
으로/로	으로서/토서	으로써/토써	
이락/락			

이를 처리해 주는 모든 명령은 각각 다음과 같다.

<code>\은 \는</code>	<code>\이 \가</code>	<code>\을 \를</code>	<code>\와 \과</code>
<code>\으로 \로</code>	<code>\으로서 \토서</code>	<code>\으로써 \토써</code>	<code>\이락 \락</code>

이 가운데 ‘`\락`, `\이락`’는 `ko.TeX/utf`와의 호환을 위하여 마련된 명령이며, 아래 설명하는 지정사 탈락을 부분적으로 지원하는 것이다.

매개 모음 “으”의 삽입은 명령 “`{_}`”를 통해 구현다. 명령 이름은 “으”의 모음 “`_`”이다. 예를 들어 “`\토`”의 정의는 “`{_}토`”이다. 매개 모음의 삽입은 동사의 굴절에서도 나타나지만 이를 자동으로 처리해야 할 경우는 없을 것으로 보인다. 그 이유는 동사가 굴절을 하지만 어간이 없는 동사가 없기 때문에 매개 모음의 앞에는 정해진 동사 어간이 있기 때문이다. 한글 라텍에서 제공되지 않은 경우의 매개 모음의 처리는 “`{_}`”를 삽입하여 해결할 수 있다.

지정사 “이”의 탈락은⁵ “교수(이) 시니 선생님이시니?” 혹은 “라텍이나 워드(이) 나가 관건이다” 등과 같이 많은 경우가 있다. 경우의 수가 많은 이유는 지정사가 동사의 한 하위 부류로서 다양한 굴절을 보이기 때문이다. 따라서 탈락 가능한 지정사에는 다음의 일반적인 명령을 사용한다. 명령의 이름에 사용된 문자는 “이”의 모음 “ㅣ”이다.

`{\ㅣ}`

utf

ko_{TeX}/utf에서는 ‘\ㅣ’와 ‘\—’를 채택하지 않고 있다. 그러므로 호환가능한 문서를 작성할 때 이 부분에 주의를 요한다. 그 대신 “지정사 탈락”에 대비하기 위하여 ‘이라(고/서/...)’ 단 한 경우만을 지원하며, 이것은 ‘\탁’ ‘\이탁’로 구현하고 있다.

`\cite` 명령으로 문헌 인용을 할 때 `plain.bst` 모양새 파일을 쓰면 인용의 이름표가 숫자로 표현되고 `halpha.bst` 모양새 파일을 쓰면 인용의 이름표가 문자열로 표현된다.⁶ 후자의 경우, 한글 라텍은 음운 변화 처리 기능을 쓰는데 있어서 문자열에 다음과 같은 제한을 두고 있다.

1. 문자열은 두 바이트 이상이어야 한다. (`halpha.bst`를 쓰면 문자열은 보통 다섯 글자 이상이 된다. 우리말은 한 글자가 두 바이트 혹은 세 바이트로 구성되므로 우리말 한 글자는 문제의 대상에서 제외된다.)

옳음: `\bibitem[HKD]{홍길동}...`

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

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

2. 문자열의 마지막 음절은 한글과 한자, 우리말 자소, 숫자 그리고 영문자로 구성된다. 즉, 자소를 제외한 상징 기호는 자동 음운 변화 처리에서 고려의 대상이 되지 않는다.⁷ 다시 말하자면, 자동 음운 변화 처리 기능은 작동하나 그 결과는 문법에 맞지 않을 수 있다. 이형태의 결정은 실질 형태소의 마지막 음절이 ‘ㄹ’이 아닌 종성인 경우로 간주된다.
3. 문자열의 마지막 글자가 영문자일 경우에, 자동 음운 변화 처리 기능은 낱말의 음가를 고려하지 않고 마지막 문자의 우리말 음가로 이형태를 결정한다.⁸

⁵ 지정사 “이다”는 학자에 따라 조사로 취급하기도 하고 상태 동사의 하나로 분류하기도 한다. 여기서 국어학적 측면은 논외로 한다.

⁶ `alpha.bst`를 쓰면 우리말이 깨진다.

⁷ 상징 기호가 인용의 이름표로 나올 일이 없을 것으로 생각한다.

⁸ 모든 영어 낱말의 우리말 음가를 올바르게 분석하기는 매우 힘들다.

옴음: <code>\bibitem[set]{홍길동}...</code>	← 영문자 이름표 <code>set</code> 는 마지막 문자 <code>t</code> 에 의해 이형태를 결정한다. <code>t</code> 의 우리말 음가는 중성으로 끝나는 음절이다. <code>set</code> 의 우리말 음가는 세트로서 역시 마지막 음절이 중성으로 끝난다. 문헌 인용의 결과는 <code>[set]</code> 토큰이다.
틀림: <code>\bibitem[ping]{홍길동}...</code>	← 영문자 이름표 <code>ping</code> 은 마지막 문자 <code>g</code> 가 이형태를 결정된다. <code>g</code> 의 우리말 음가는 중성으로 끝나는 음절이다. 그러나 <code>ping</code> 의 우리말 음가는 핑으로서 중성으로 끝나는 음절이다. 문헌 인용의 결과는 <code>[ping]</code> 토큰이다.

옴음: <code>\index{절 see{첫장~첫절}}</code> <code>\index{절 see{첫장\ 첫절}}</code>	← ‘~’나 ‘\’는 모두 빈 칸을 의미한다. ‘~’는 앞뒤의 단어를 같은 줄에 출력하도록 하고 ‘\’는 다른 기능없이 빈 칸만 출력한다.
틀림: <code>\index{절 see{첫장 첫절}}</code>	

출력: 첫장 첫절을 참고

그림 6.2: 색인 작성

`halpha.bst`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 경우) 세 번째의 제한은 무시될 수 있다. 보통의 경우에도 `\cite`의 이름표(`label`)는 일련 번호이거나 (`plain.bst`) 저자이름 + 책출판년도를 약식으로 표현한다 (`halpha.bst`). 그러므로 보통은 발생하기 힘든 경우에 해당한다. 그러나 참고 문헌 작성 시에는 이런 사실을 알고 있으므로써, 발생할 가능성이 있는 오류를 사전에 방지하는 것이 좋다.

색인 작성 시에 `makeidx.sty`을 쓰면 `\see` 명령이 정의된다. 이 명령을 쓸 때는 `\see`의 변수가 빈 칸을 포함하게 되면 자동 음운 변화 처리 기능에 의해 오류가 발생한다. 빈칸 대신에 “~”나 “\”를 쓰면 문제가 발생하지 않는다. 예: 그림 6.2

`\pagename`, `\chaptername`과 같이 우리말화 한 한글 이름들도 자동 음운 변화 처리를 할 수 있다. 그러므로 `\pagename\` 가라고 쓰면 “페이지가”라고 출력된다.

2.7 우리말 이름

한글 라텍스에서 사용되는 각종 이름들을 일괄적으로 `\ksnamedef` 모든 명령에 의해 변경될 수 있도록 하였다. 이러한 “한글 이름”을 사용하려면 `[hangul]` 옵션을 지정하여야 한다.

예를 들어, 라텍스에서 사용되는 ‘참고 문헌’의 이름 ‘Reference’는 `\refname`에 지정되어 있다. 이를 우리말화 하는 방법은 다음과 같다.

```
\ksnamedef{refname}{참고~문헌}
```

utf

ko,TeX/utf 에서도 `[hangul]` 옵션을 준 경우라면 이 명령을 사용할 수 있다.

표 6.2에는 한글 라텍스에서 한글화한 “라텍 이름의 한글 및 한자화”가 나열되어 있다. 이 한글화는 `\begin{document}` 앞에서 위에 예를 든 것처럼 `\ksnamedef`로 재정의하면 원하는대로 변경된다.

`\ksTHE`는 라텍스에 정의되어 있지 않고, 한글 라텍스에서 단원의 숫자 판짜기에 사용되도록 도입된 이름 모든 명령이다. 이는 라텍 이름을 한글/한자화 함으로써 발생 하는 문제점들을 해결하기 위해 사용된다. 예를 들어 `\part`의 일련 번호는 단원 숫자의 한글화로 인해 “편 I”과 같이 나오며 한글 라텍에서는 `\ksTHE`를 도입하여 “제 I 편”과 같은 식으로 짜여지도록 하고 있다.

한글 라텍에서는 `\part`와 `\chapter`, `\appendix`, `\section`까지를 위와 같이 한글화하고 있으며, 각각 다음과 같이 설정되어 있다.

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> <code>\thechapter</code>	article book/report
<code>\section</code>	<code>\ksTHE~</code>	<code>\thesection</code>	<code>~\sectionname</code>

이를 사용자의 편의에 따라 재설정하기 위해서는 다음의 모든 명령을 사용한다.

```
\kscntformat{단원이름}{앞}{뒤}
```

예를 들어 `\chapter`를 “첫째 마당”과 같은 식으로 짜기 위해서는 전문(preamble)—`\begin{document}` 앞—에서 다음과 같이 지정한다.

라텍명령	한글 라텍 (한글)	한글 라텍 (漢字)
today	1994 년 3 월 6 일	1994年 3月 6日
enclname	동봉물	同封物
ccname	사본	寫本
headtoname	받는이	受信人
seenname	\을 참고	\을 參考
contentsname	차~레	目~次
listfigurename	그림~ 차례	그림~ 目次
listtablename	표~ 차례	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	용어집	語~ 彙
pagename	페이지	쪽
figurename	그림	그림

표 6.2: 라텍 이름의 한글 및 한자화

```

\renewcommand{\thechapter}{\Hnum{chapter}}
\ksnamedef{chaptername}{막당}
\kscntformat{chapter}{}{~\chaptername}

```

2.8 그외의 우리말화

우리말의 분철은 불임표가 사용되지 않으므로 영어권 언어와는 다른 특별한 처리 방법이 요구된다.

영어 문서에서 단원 제목에 이은 첫 문단에서는 들여쓰기를 하지 않는다. 그러나 한글 문서에서는 모든 문단에 들여쓰기가 적용되는 것으로 보인다. (이는 한글 문서 작성의 표준이 규정된 것으로 보이지 않으나 한국의 출판 서적을 볼 때 그렇게 판단

된다.) 첫 문단 들여쓰기를 하는 국가는 한국 뿐만은 아닌 것으로 보이며, 텍 운영 체제에서 `indentfirst` 꾸러미를 사용하여 이를 해결하도록 되어 있다. 즉, 문서의 전문 영역에서

```
\usepackage{indentfirst}
```

를 지정하면 첫 문단 들여쓰기가 쉽게 이루어진다.

또한 들여쓰기의 정도에 있어서도 영어 문서와 우리말 문서는 차이가 있는 것으로 보인다. (역시 정해진 표준이 있는 것으로 보이지 않고 출판사에서 관습적으로 들여쓰기의 양을 정해 쓰는 것으로 보인다.) 들여쓰기의 양은 한글 글자 두개의 넓이 즉 전각 2배각을 쓰는 것이 적당해 보인다. 이는 많은 한국어 서적에서 보이는 전각 들여쓰기보다 큰 값이지만 (이중운, 1991)과 KS A 0104에서 채택되고 있는 값이고 우리말 문서에 적당하게 보인다. 이 들여쓰기의 양은 다음과 같이 설정할 수 있다.

```
\parindent=2em
```

제 3 절 한글화

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

문자 범주를 기대하는 명령의 변수가 한글일 때 라텍은 오류를 유발한다. 특히 `\label`이나 `\cite` 명령 등의 변수가 한글일 때가 그러하다. 한글 라텍에서는 표준 문서 양식에서 사용되는 이와 같은 명령들을 수정하여서 한글 라벨을 사용할 수 있도록 했다. 그러나 다른 패키지에서 이런 명령들을 재정의하면 또다시 문제가 발생할 수 있다.

문제가 되는 것 중에 `hyperref` 패키지가 있다. pdf 제작에 있어서는 거의 필수적인 이 패키지가 한글 라벨 이름을 받아들이지 않는다. pdf를 제작하거나 `ko.TEX/utf`와의 호환을 고려한다면 되도록 한글 라벨 이름을 쓰지 않는 것이 좋다.

utf

`showkeys` 패키지는 `\label`, `\ref`, `\pageref`, `\cite`, `\bibitem` 명령을 수정하여 내부적으로 사용되는 표지를 출력하는 기능을 수행한다. 한글 라텍스에서는 이 패키지를 한글화하여 `showhkeys.sty`의 이름으로 제공한다.

참고 문헌의 인용에 사용되는 표지를 `showkeys` 패키지와는 다른 방식으로 출력하는 `showtags` 패키지를 한글화한 파일은 `showhtags.sty`이다.

제 IV 편

한글 문서 작성

제 7 장 한글 문서 서식

제 1 절 각주 형식

1.1 hangulfn의 각주 판짜기

이 패키지는 ko.TeX/euc 에서는 `hangulfn`, ko.TeX/utf 에서는 `dhucsfn`이라는 이름으로 되어 있으며, 하는 일은 동일하다. 두 패키지 모두 영문자 옵션을 받아들이지만 `dhucsfn`은 한글 옵션을 처리하지 못한다. 영문자 옵션을 주로 사용할 것을 권장한다.

패키지의 사용법은 다음과 같다.

```
\usepackage[선택사항]{dhucsfn}
```

또는

```
\usepackage[선택사항]{hangulfn}
```

선택사항으로는 각주 번호의 판짜기 방식을 지정하는 **첨자**와 **괄호**가 있는데 각각 다음과 같이 짜여진다.

- **첨자(superscript)**: ¹ 각주문
- **괄호(parenthesis)**: 1) 각주문

첨자의 경우에는 각주 번호와 각주문 사이가 붙고 괄호의 경우에는 소괄호와 각주문 사이에 구분 공간이 있다. 애초값은 첨자이다.

각주 판면의 판짜기를 지정하는 선택사항의 종류는 다음과 같다.

- **다항이어쓰기(multipara)**: 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

- **단순 이어쓰기(para):** “다행이어쓰기”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

- **내어쓰기(hang):** 각각의 각주는 새 행에서 시작한다. 각주 번호가 각주문의 왼쪽으로 내어 써 지고, 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.

- **왼쪽 맞춤(leftflush):** 각각의 각주는 새 행에서 시작한다. 각주의 첫 줄이 왼쪽맞춤으로 짜진다.

12) 각주 번호와 각주문이 들여쓰기나 내어쓰기에 의해 구분되지 않고 전체가 하나로 왼쪽맞춤 되는 판짜기 방식이다. 각주 번호와 각주문의 사이에는 구분 공간이 삽입된다. 구분 공간은 첨자의 경우 0pt이고 괄호의 경우 공간 문자이다.

- **들여쓰기(indent):** 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 들여쓰기 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.

- **들여왼쪽 맞춤(leftflushindent):** “왼쪽 맞춤”과 같은 방식이다. 전체 각주면이 본문으로부터 오른쪽으로 2배각 들여 써진다.

12) 각주 번호와 각주문이 들여쓰기나 내어쓰기에 의해 구분되지 않고 전체가 하나로 왼쪽 맞춤 되는 판짜기 방식이다. 전체 각주면이 본문의 왼쪽맞춤 위치에서 2배각 들여진다.

- **들여내어쓰기 (hangpar):** “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 (varhangpar):** “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.

1.2 각주 문단 모양 관련

한글 문서는 기본 행간을 영문 문서에 비하여 넓혀 잡기 때문에 각주 사이의 간격이 흐트러져서 모양이 일그러지는 경우가 생긴다. 즉, 한글 문서에서는 각주 사이의 간격을 적당하게 재조절해주어야 한다. 이에 대해서 `dhucs-setspace` 패키지(utf) 또는 `hsetspace` 패키지(euc)가 약간의 해결책을 제시해주시만, 각주 내의 행간을 재조절하는 특징이 있다. 일반적인 문서에서 각주 행간을 본문 행간과 동일하게 할 것인가 아니면 좁은 행간을 쓸 것인가는 전적으로 선택의 문제이지만, `\footnotesep` 길이 값을 저자가 직접 설정해줄 필요가 있다는 점을 기억해두자.

각주 문단이 다음 페이지로 넘어가는 문제와 관련하여, \LaTeX 에서는 각주 번호가 찍힌 페이지보다 앞에 각주가 먼저 나오는 경우가 없다는 것을 알아두어야 한다. 이 때문에 각주 위치잡기를 위해서 페이지의 하단이 비는 경우도 생기는데, 이를 해결하는 여러 가지 방법이 \LaTeX 관련 팁으로 제시되고 있다. 그 가운데 하나는 `bigfoot` 패키지를 사용하는 것이다. `bigfoot` 패키지나 `footmisc` 패키지 등의 사용법을 익혀두면 문서 작성에서 도움을 얻는 경우가 많다.

제 2 절 자간, 행간, 단어간격

2.1 자간

자간을 처리하는 데는 있어서 알아두어야 할 것은 다음과 같다.

ko_{TeX}/euc에서 자간은 `\hangulskip`이라는 길이값을 조절하는 것으로 가능하다. 예를 들면,

```
\hangulskip=0pt
```

ko_{TeX}/utf에서는 비슷한 기능을 `\setInterHangulSkip`이라는 명령을 통하여 지정할 수 있다.

```
\setInterHangulSkip{0pt}
```

또한 폰트마다 각각 다른 자간값을 폰트 속성으로 부여해두었으므로, 위의 명령을 지정했다 하더라도 `hfontspec`을 불러들이면 모두 폰트 속성값으로 대체된다. 따라서 위의 명령은 그다지 유용하지 못하고 사용자가 문서에서 활용할 만한 것이 아니다. 실제 자간은 판면의 모양을 심하게 일그러뜨리므로 되도록 사용자 수준에서 그 조절은 삼가야 할 것이다.

그럼에도 불구하고, 자간을 조절할 수 있는 유틸리티 패키지가 제공된다. 이것은 `hlatex-interword(euc)` 또는 `dhucs-interword(utf)`라는 패키지의 기능 중의 하나이다. `\interhchar` 명령을 이용한다.

```
\usepackage{dhucs-interword} % or, hlatex-interword (euc)
\interhchar{0pt}
```

ko_{TeX}/utf에서 `\setInterHangulSkip`과 `\interhchar`는 사실상 동일한 명령이다.

`dhucs-interword` 패키지 등을 썸에 있어 주의할 점은, `[default]` 옵션으로 로드하면 기본 자간이 0pt로 리셋된다는 점이다. 그러므로 폰트 속성 자간값을 적용하려면 `dhucs-interword` 이후에 `\usehangulfontspec` 명령을 호출하는 것이 좋다.

```
\usepackage[default]{dhucs-interword}
\usehangulfontspec{ut}
```

`dhucs-interword` 패키지가 의도대로 동작하려면 ko_{TeX}/utf에서 `finemath`가 활성화되어 있어야 한다.

2.2 단어간격

단어 간격과 관련된 \TeX 명령은 `\spaceskip`이다. 그리고 `nonfrench spacing`에서 `\xspaceskip`은 마침표 뒤의 여분 공백의 크기를 지정한다. `ko \TeX /euc`는 언제나 `nonfrench`로만 동작하지만 `ko \TeX /utf`는 `nonfrench` 옵션을 지정해야 한다는 사실을 기억해두자.

`dhucs-interword` 패키지 또는 `hlatex-interword` 패키지는 단어간격의 기본 값을 설정하거나 바꿀 수 있게 하는데, `\interhword` 명령이 그것이다. 다만 이 두 경우 사용법이 조금 다르다는 사실에 주의하자.

dhucs-interword

```
\usepackage{dhucs-interword}
\interhword[.6]{.475}{.1}{.1}
```

여기에서 옵션 인자는 `\xspaceskip` 값을 의미하며, `french spacing`에서는 무의미하다. 이 값들은 문서 기본 폰트 사이즈의 승수값을 의미한다. 즉, `.475`라는 것은 $0.475 \times 10\text{pt} = 4.75\text{pt}$ 가 되는 것이다. `[12pt]` 옵션의 문서라면 이 값이 달라진다.

hlatex-interword

```
\usepackage{hlatex-interword}
\interhword{.475em}{.1em}{.1em}
```

이 경우에는 길이 단위를 지정해주어야 한다.

두 경우 모두, 첫번째 인자와 두번째, 세번째 인자 사이는 다음과 같은 규칙으로 적용된다.

`<first> plus <second> minus <third>`

자간과 단어 간격을 설정하는 `hlatex-interword` 패키지와 `dhucs-interword` 패키지에서 사용할 수 있는 옵션과 매크로는 다음과 같다.

[default] 옵션 한글 문서에 적당한 자간과 어간을 설정한다. 영문자의 자간 및 어간보다 조금 넉넉한 정도이다.

[HWP] 옵션 아래아한글 97의 기본 자간 및 어간 설정을 흉내낸 것이다. **[default]**보다 어간이 더 넓다.

`\interhword` 매크로 어간을 임의로 설정하게 한다.

`\interhchar` 매크로 자간을 임의로 설정하게 한다. 한 개의 인자를 취하는데 반드시 길이단위를 붙여주어야 한다. `\interhchar{0pt}`는 자간을 0 point로 만드는 것이다. 다만, 이 방식에 의한 자간의 변경은 그다지 권장하지 않는다. 왜냐하면 자간은 폰트 자체의 속성으로 간주하여 `hfontspec`에 의해 제어되는 것이 바람직하다고 보기 때문이다. `dhucs-interword` 패키지는 단어 간격의 제어에만 활용되는 것이 좋으리라고 생각한다.

2.3 행간

행간은 `\baselinestretch`나 `\linespread`를 이용하여 간단하게 제어할 수 있지만 좀더 복잡한 경우도 없지는 않다. 예를 들면 문서 일부에 별도의 행간을 적용한다든가...

행간을 제어하기 위하여 제공되는 패키지는 `hsetspace` 또는 `dhucs-setspace` 패키지이다. 이 패키지는 `setspace` 패키지를 한글화한 것이다.

[`hangul`] 옵션을 활성화하거나 이 패키지를 로드하면 기본 행간이 1.333으로 변경된다. 그리고 `setspace` 패키지의 모든 기능을 다 쓸 수 있다. 이 패키지의 사용법은 해당 패키지 문서를 참고하라.

```
\usepackage[hangul]{dhucs-setspace}
```

`setspace` 패키지의 독특한 특징 중 하나가, 행간을 늘리더라도 `floats`(그림과 표), `footnote` 안의 행간은 여전히 1.0으로 고정된다는 것이다. 한글화 과정에서 이 개념을 확대하여, 넓은 행간과 좁은 행간 두 종류로 나누고 좁은 행간을 `footnote`, `floats`, `quote`, `verbatim` 등에 적용한다. 이를 설정하는 명령이 `\SetHangulspace`이다.

```
\SetHangulspace{1.333}{1.2}
```

이 패키지는 좀 복잡한 옵션을 가지고 있는데 이를 설명하면 다음과 같다.

`hsetspace` 패키지와 `dhucs-setspace` 패키지는 `setspace` 패키지를 불러온다. 즉, `singlespace`, `doublespace`, `onehalfspace` 등의 환경과 `\singlespacing` 등의 선언을 사용할 수 있다. 한글화를 위해서 추가된 옵션과 매크로만을 소개하면 다음과 같으며, 실제 사용법은 `setspace` 패키지 매뉴얼을 참고하라.

[`nofloatspacing`] 옵션 `figure`, `table` 등의 `float` 내에서 간격을 줄이는 기능을 끈다.

[`noquotespacing`] 옵션 `quote` 환경 안에서 간격을 줄이는 기능을 끈다.

[hangul] 옵션 한글화 매크로인 `\SetHangulspace` 명령을 쓸 수 있게 하고 행간을 한글 문서 기본값으로 설정한다.

[adjustverbatim] 옵션 verbatim 환경 안에서 좁은 간격을 적용한다.

[adjustfootnotesep] 옵션 각주는 기본적으로 좁은 간격으로 된다. 그러나 각주 간 간격은 자동으로 맞추어지지 않는데 이것을 조금 보정하도록 한 것이다. 행간을 임의로 설정하려 할 때는 무의미하고 기본값에 대응시킨 것이다.

`\SetHangulspace` 매크로 두 개의 인자를 취하여 첫번째 것을 기본 행간, 두번째 것을 좁은 행간 간격으로 한다. 인자는 모두 stretch 값이어야 한다. 예를 들면 `\SetHangulspace{1.3}{1.1}`과 같은 방법으로 사용하며, preamble에서만 사용한다. ‘좁은 행간’은 float, 각주 등에 사용되는 행간을 말한다.

euc 버전과 utf 버전은 동일하게 동작한다.

제 3 절 ko.TeX/euc에서의 밑줄

45 페이지의 4.4.4 절에서 말한 바와 같이 ko.TeX/utf와는 달리 ko.TeX/euc는 ulem 패키지와 약간의 충돌이 있다. 부분적이거나 이 문제를 피하기 위해 myulem 패키지라는 작은 패키지가 제공된다. 다음과 같이 사용한다.

```
\usepackage{myulem}
```

밑줄 명령들 앞에는 h를 붙여서 쓴다.

```
\huline{가나다}
```

```
\huwave{가나다}
```

```
\hsout{가나다}
```

```
\hxout{가나다}
```

```
\huuline{가나다}
```

결과는 다음과 같다.

가나다 가나다 가나다 가나다 가나다

다만, ko.TeX/euc에서는 ko.TeX/utf와는 달리, 밑줄 명령 안에서는 단어 단위로만 행나눔이 이루어진다. ko.TeX/utf는 밑줄 명령 안이라도 글자 단위로 행나눔이 가능하다. 여러 가지 면에서 ko.TeX/utf의 ulem 패키지를 사용하는 방식이 더 낫다.

제 4 절 장절표제

ko_{TeX}은 [hangu1] 옵션을 두고 있다. 이 옵션을 활성화하면 두 가지 중요한 변화가 일어난다.¹

1. 한글식 이름(names)을 사용한다. 그래서 Figure와 같은 표제 이름이 ‘그림’으로 바뀐다.
2. 한글식 장절표제 모양이 활성화된다. 예를 들면 “Chapter 1”이 아니라 “제 1 장” 형식으로 식자한다.

그러나 이것만으로는 충분하지 못하다. 장절표제의 형식을 바꾸는 많은 패키지들이 있는데 이것과 한글 장절표제 형식이 일치하지 않아서 생기는 문제들이 많이 있다. 모든 관련 패키지를 다 지원할 수는 없지만 sectsty 패키지는 이를 한글화해 두었다. hsectsty 패키지(euc) 또는 dhucs-sectsty 패키지(utf)이 그것이다. 이 패키지의 장절표제 설정 명령과 함께 한글식 장절표제 형식을 유지하도록 해준다. sectsty 패키지의 사용법은 해당 패키지 문서를 참고하라.

hsectsty 패키지와 dhucs-sectsty 패키지는 [ensec] 옵션을 줄 수 있다. 이렇게 하면 한글 서식을 적용한 경우에도 절 제목에 한해서 영문 문서와 같이 “제”와 “절” 없이 숫자만 찍힌다.

oblivoir를 사용한다면 memoir의 간편한 방식을 사용하여 더 쉽게 장절표제 형식을 사용자화할 수 있다.

제 5 절 문장부호

5.1 우리말 문장부호의 사용법

우리말 문서에서 사용되는 문장 부호는 라텍스에서 제공되는 문장 부호가 사용될 수 있다. 그러나 특수한 경우에는 우리말 문자 체계에서 제공하는 상징 문자를 사용해야 하고 경우에 따라서는 특별한 명령으로 구현되어야 한다. (한글연구회, 1992)의 부록에서는 우리말 문서에서 다음과 같은 문장 부호의 사용법을 규정하고 있다.

5.1.1 마침표

- 온점(.), 고리점(。)

가로쓰기에는 온점, 세로쓰기에는 고리점을 쓴다.

¹ko_{TeX}/utf에서는 추가적인 번호 형식, 추가적인 방점 명령 등을 더 사용할 수 있게 된다.

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!

5.1.2 쉼표

- 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.

- 가운뎃점 (·)

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 윗놀이를 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 때에 쓰기도 한다.

영문자(/)를 입력한다.

착한 사람 / 악한 사람 맞닥뜨리다 / 맞닥트리다

5.1.3 따옴표

- 큰따옴표 (“”), 겹낫표 (『 』)

가로쓰기에는 큰따옴표, 세로쓰기에는 겹낫표를 쓴다. 대화, 인용, 특별 어구 따위를 나타낸다.

큰따옴표는 라텍의 입력 방식에 따라 작은따옴표를 두 번 (“”) 입력하여 사용하고 겹낫표는 우리말 상징 부호 (『 』)를 입력하여 사용한다. (우리말 상징 부호의 겹낫표는 시계 반대 방향으로 90° 회전된 형태이다. 아마도 가로쓰기에서 사용될 수 있도록 변화된 것으로 보인다.)

“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 부호의 낫표는 시계 반대 방향으로 90° 회전된 형태이다. 아마도 가로쓰기에서 사용될 수 있도록 변화된 것으로 보인다.)

지금 필요한 것은 ‘지식’이 아니라 ‘실천’입니다.

5.1.4 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 꺾음 정도가 다르므로 ([]) 우리말 상징 부호를 입력하여 사용할 수 있다.

명령에 있어서의 불확실[단호(斷乎)하지 못함]은 복종에 있어서의 불확실[모호(模糊)함]을 낳는다.

5.1.5 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, 말씀드렸다가—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 사용한다.

겨울-나그네 빛-에너지

- 물결표 (~)

‘내지’라는 뜻에 혹은 어떤 말의 앞이나 뒤에 들어갈 말 대신 쓴다.

라텍 명령 `\textasciitilde`를 입력하거나 우리말 상징 부호(~)를 입력하여 사용한다. 아래의 왼쪽 예문에서와 같이 라텍 명령에 사용되는 물결표는 액센트 문자를 만드는 역할을 하므로 우리말의 물결표로는 부적당하게 보인다.

9월 15 일~9월 25 일 미술~

5.1.6 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.

5.1.7 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 둠을 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 은 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.

5.2 문장부호에 관련된 문제

ko.T_EX/utf는 [finemath] 옵션으로 물음표와 느낌표, 그리고 온점 마침표의 수직 위치를 조절하는 기능이 있다. 이것은 영문 폰트의 아스키 문장부호를 가져다 쓰기 때 문에 불가피한 것이었다. 또한 수평 간격도 일부 조절해준다.

이 기능은 ko.T_EX/euc에는 적용되지 않는다.

제 8 장 색인과 문헌목록

제 1 절 ko.T_EX/euc 의 hbibtex 과 halpha

1.1 우리말 문헌 인용

라텍에서 문헌을 인용하기 위해서는 `\cite` 명령으로 참고 문헌 표지를 지정한다. 참고 문헌 표지는 thebibliography 환경 내에서 `\bibitem`으로 지정되는 참고 문헌 항목에 주어지거나 BibT_EX을 이용하는 참고 문헌 데이터 파일 .bib의 목록에서 주어진다. 참고 문헌 데이터 파일을 사용하면 정해진 양식으로 참고 문헌 정보를 .bib 파일에 수집하여 묶으로써 어느 문서에서든지 그 정보를 꺼내 쓸 수 있고 참고 문헌 양식 파일을 지정하여 다양하고 일관된 참고 문헌 목록을 작성할 수 있다.

참고 문헌 양식 파일은 개별 언어에 맞게 수정되어야 할 부분이 많다. alpha.bst의 경우에는 저자나 편집자와 같은 이름을 약자로 바꾸기도 하는데, 이 때에는 한글 이름에서 오류가 발생하게 된다. 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 때문이다. 한글 라텍은 alpha.bst를 한글화한 halpha.bst를 제공한다. 예를 들면 halpha.bst를 사용하여 참고 문헌 목록을 작성할 수 있다.

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

위에서는 참고 문헌 양식으로 halpha.bst를 사용하도록 지정하고 참고 문헌 데이터 파일로 hlguide.bib, texbook1.bib, texbook3.bib를 지정한 것이다.

BibT_EX을 이용하여 한글 참고 문헌을 만들기 위해서는 라텍이 8비트 문자를 보조 파일 (확장자가 .aux)에 출력할 수 있어야 한다. 적어도 teT_EX-3.0 (web2c-7.5.4) 버전에서는 tcx 파일을 이용하여 latex.fmt 파일을 만듦으로써 8비트 출력이 가능하도록 되어 있다. 그러나 lambda.fmt은 tcx 파일을 사용하지 않을 뿐만 아니라 프로그램 원천에서도 EUC (Extended Unix Code) 문자가 T_EX의 16진법 표기로 출력되도록 프로그래밍 되어 있다. 여기서 문제가 발생하는 부분은 참고 문헌 표지가 한글로 되어 있을 경우이다. 예를 들어 참고 문헌 표지가 “홍91”일 경우에 “`\cite{홍91}`”

과 같이 인용을 했을 경우에 Λ 는 보조 파일에 “\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 찾아낸다. 그러나 참고 문헌 데이터 파일에 기록된 표지는 “홍 91”이므로 이 항목을 찾아 내지 못한다. 따라서 한글 참고 문헌 데이터 파일을 사용하기 위해서는 보조 파일을 8비트 문자로 변환할 필요가 있고, 한글 라텍스에서는 C 프로그래밍 언어로 짠 hbibtex을 제공하여 BibTeX을 사용할 수 있도록 하였다.

hbibtex의 사용법은 bibtex의 사용법과 동일하다. 라텍스를 사용할 경우나 참고 문헌 표지에 한글을 전혀 쓰지 않을 경우에는 hbibtex을 사용하지 않아도 된다.

1.2 참고 문헌 데이터베이스

라텍스로 문서의 판짜기에 익숙해진 후에는 참고 문헌 목록의 관리에 관심을 가질 수 있다. 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 된다. 또한 참고 문헌 데이터베이스는 특정 분야에 독립적인 형식으로 구축되고 이 데이터베이스를 이용하는 사용자가 필요한 정보를 원하는 모양으로 출력할 수 있다.

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

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

```
@<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 다르다.

1.2.1 문헌 유형

대부분의 참고 문헌 양식 파일에서 사용되는 문헌의 유형과 필수/선택 사항은 다음과 같으며 필수/선택 사항이 아닌 사항은 `bibtex`에 의해 무시된다.

- **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, title, 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/계간과 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

1.2.2 기재 사항

각 기재 사항의 내용은 다음과 같다.

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

1.2.3 저자 및 편집자 이름 형식

author나 **editor**의 영어 이름은 다음의 두 가지 형식으로 입력된다.

"<이름> <성>"

"<성>, <이름>"

그러나 한국어 이름은 공백 문자가 없이 다음과 같은 형식으로 입력된다.

<성><이름>

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

이름에 사용되는 기능어에는 **and**와 **others**가 있다. **and**는 여러 이름을 나열할 때 사용되고 **others**는 여러 이름을 축약할 때 사용된다. 따라서 이름 기재 내용이 **and others**로 끝날 때에 이는 “_외”(영어의 경우는 “*et al.*”)로 출력된다.

제 2 절 색인

2.1 ko.TeX/utf의 색인 처리

ko.TeX/utf는 kotex.ist라는 색인 스타일 파일과 komkindex라는 makeindex 유틸리티를 제공한다. makeidx 패키지에 의해 만들어지는 foo.idx 파일을 색인 환경으로 변환해주는 역할을 한다.

이것은 다음과 같은 형식으로 쓸 수 있다.

```
$ komkindex(.pl) -s kotex foo
```

이 명령은 foo.idx에서 foo.ind 파일을 만들어내며, 한글 자모를 기준으로 정렬하여 준다. makeindex 옵션을 그대로 쓸 수 있다.

2.2 ko.TeX/euc의 우리말 색인 처리

ko.TeX/euc를 위한 색인 처리 유틸리티는 hmakeindex 이고, 이것은

```
$ komkindex(.pl) -euc
```

와 같이 호출한다. 추가 옵션은 이 뒤에 이어서 쓴다.

```
$ komkindex(.pl) -euc -s hind foo
```

이 이후 hmakeindex를 부르는 예들은 모두 komkindex -euc로 실행할 수 있다.

hmakeindex는 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 삽입한다. \hindexhead의 애초값은 \indexspace이며 줄간격을 늘리는 역할을 한다. 만일 색인의 머릿제목을 “『가』, 『나』, 『다』, 『라』, ...”로 표시하고자 한다면 다음과 같이 라텍 문서에서 \hindexhead를 재정의할 수 있다.

```
\renewcommand\hindexhead[1]{\indexspace
  {\bfseries
    『 \ifcase#1 심벌\or 가\or 까\or 나\or 다\or 따\or 락\or
      막\or 박\or 빠\or 사\or 싸\or 아\or 자\or 짜\or
      차\or 카\or 타\or 파\or 하\or 한자\fi 』}
  \nopagebreak
}
```


`hmakeindex` 프로그램은 C 언어로 작성되었으므로 운영 체제에서 기계어로 번역한 후 설치하여야 한다. 윈도우즈 운영 체제에서는 미리 번역한 `hmakeindex.exe`를 설치하면 된다. 색인작성은 다음과 같이 한다.

```
komkindex -euc <문서이름>
```

한글 색인 양식 파일은 여전히 존재하며 색인과 쪽 번호 사이를 점선으로 채우는 기능만 수행한다. 이 모양새 파일 `hind.ist`를 지정하기 위해서는 `komkindex -euc`의 추가 선택으로 `-s hind`를 지정한다.

```
komkindex -euc -s hind <문서이름>
```

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

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

제 9 장 한글 트루타입 폰트의 사용

ko₂T₂X 유틸리티로서 `ttf2kotexfont.pl` 이 제공된다. 이 유틸리티는 한글 트루타입 폰트를 사용가능하도록 `tfm` 을 추출하고 설정을 행해주는 역할을 한다.

트루타입 폰트 이용법 전반에 관한 사항은 “한글 라텍에서 트루타입 폰트 사용하기”((김강수 & 주철, 2004:2007)) 라는 글을 참고하라. 여기서는 간단한 예를 통하여 트루타입 폰트를 문서에서 사용하는 과정을 설명하고자 한다.

제 1 절 ttf2kotexfont 사용법

적당한 폰트 두 개를 선정하자. 공평하게 하기 위해 유니코드 인코딩 폰트인 “휴먼편지체”와 완성형 인코딩 폰트인 “양재난초” 두 개를 선택. 파일 이름은 `hmfmpyun.ttf` 와 `YNCH05.TTF` 이다.¹

이 글꼴이 시스템에 설치되어 있다면, 별다른 조치 없이 바로 테스트 가능. 만약 없다면 어디선가 구해서 테스트를 위한 폴더에 넣어준다.

`test.conf` 파일을 다음과 같이 작성한다.

```
FOUNDRY:my
FONTmj: f=pj m=hmfmpyun.ttf
FONTnc:      m=YNCH05.TTF
```

`config` 파일의 쓰는 규칙은 위와 같다. 맨 처음에 `FOUNDRY` 를 지정해주어야 하는데 이것이 글꼴 가족 이름의 첫 두 글자를 이룬다. `FONTmj` 행은 반드시 있어야 하며, 이 이외의 폰트 명칭은 두 글자로 지은 다음 그 앞에 `FONT` 를 써서 적어주면 된다. `f` 지시자는 `FONTmj` 로 만들되 이름을 `mypj` 로 지으라는 의미이며, 이 값이 없으면 `FONTxx` 의 두 글자 `xx` 를 따서 짓게 된다. 폰트 두께 옵션은 `m`, `b`, `l` 셋이 있으며 각각 `medium`, `bold`, `light` 에 해당하는데, `b` 는 `b-series` 및 `bx-series` 에 할당된다. 그러나 일반적으로 이 시리즈들을 모두 충족하는 글꼴군을 찾기란 쉬운 일이 아니다. 위의 예에서는 `m` 하나만 지정하였으므로 본문 서체인 보통 굵기 폰트만이 만들어질 것이다. 이러한 설정의 결과 `mypj` 와 `mync` 라는 새로운 글꼴이 만들어진다.

이제 다음 명령을 실행한다.

¹이 스크립트는 파일이름의 대소문자에 민감하다.

```
# tt2kotexfont -c test.conf
```

한참 tfm 등을 만드는 과정을 거치고 나서 종료되면 테스트해볼 수 있다.

```
# latex testutf
# dvipdfmx testutf
```

또는

```
# pdflatex testutf
```

이 폰트 세트를 반복해서 다른 문서에서도 사용할 생각이면 사용자의 texmf 트리 아래 설치해서 쓰면 된다. 언인스톨 방법이 별도로 제공되지 않으므로, 자신이 생각하기에 잘 설계하여 반복사용하겠다는 결심이 섰을 때, 다음 명령을 실행하자.²

```
# tt2kotexfont -c test.conf -i $HOME/texmf
```

만약 이 과정에서 \$HOME/texmf 가 “가장 먼저 읽는 texmf tree”가 아니라면³ 생성된 dvipdfmx.cfg와 tt2pk.cfg를 가장 먼저 읽는 texmf tree 아래 TDS에 맞는 위치로 옮겨놓는 것을 잊어버려서는 안된다.

```
# mktexlsr
```

일부 완성형 트루타입 중에 이상한 것이 없지는 않다(양재보람체가 이상했음). 그것은 폰트 자체의 문제라고 생각되며, 정상적인 대부분의 트루타입은 ko_{TeX}에서 모두 사용 가능하다.

제 2 절 문서에서 폰트 설정하기

tt2kotexfont 스크립트를 실행한 결과를 반영하려면 다음과 같이 한다. 사용자가 만든 임의의 폰트가 mypj와 mync라고 할 때,

ko_{TeX}/euc

```
\usepackage{myttf}
```

과 같이 선언한 이후, 본문에서 예컨대

```
\pjfamily, \ncfamily
```

²Windows/KC2006의 경우, 아래 명령 예시에서 \$HOME이라고 한 것을 %HOME%으로 해야 한다.

³KC2006의 경우와 같이

를 사용할 수 있다.⁴

ko.TeX/utf

```
\SetHangulFonts{mypj}{mync}{uttz}  
\SetHanjaFonts{utbt}{utgt}{uttz}
```

또는

```
\usepackage{hfontsel}  
\SelectHfonts{mypj,mync,*}{utbt,utgt,*}
```

주의할 것은 ttf2kotexfont 가 hfontspec 을 만들어주지는 않는다는 점이다. 기본은 글꼴 폰트 스펙을 적용(이 때 사용자가 취할 조치는 아무것도 없다)해도 나쁘지 않으며, 만약 직접 폰트 스펙을 변경하려 한다면 반드시 수작업으로 설정해야 한다.

제 3 절 pdf 문서 제작을 위한 pdf_{TeX} 사용

트루타입 폰트를 pdf_{TeX}에서 사용하려 할 때는 약간의 주의사항이 필요하다. 먼저, 기울인 글꼴, 가상 두꺼운 글꼴, c-series 장평적용 글꼴 등은 pdf_{TeX}에서 직접 처리되지 않는다.

또한 preamble에서 다음과 같은 코드를 넣어주어야 한다.

```
\usepackage{ifpdf}  
\ifpdf  
  \usepackage{dhucs-cmap}  
  \pdfmapfile{=myttf-pdftex.map}  
\fi
```

대체로 말해서, 트루타입 글꼴은 DVIPDFMx 가 매우 잘 처리한다.

⁴이 명령들은 전혀 호환되지 않는다. 나중에 다른 곳에 파일을 보내면 반드시 에러가 발생할 것이므로 주의해야 한다. 이 폰트가 설치되지 않았을 때도 이 명령에 에러가 발생하지 않도록 하는 코드를 preamble에 넣어 두는 것이 좋다.

제 10 장 plainTeX과 koTeX

by 남수진(mailto:sjnam@ktug.or.kr)

플레인텍과 관련된 koTeX 한글 매크로에는 kotexplain.tex과 hangulcweb.tex 두 개의 파일이 있다. koTeX의 plainTeX 지원은 오직 유니코드/UTF-8 인코딩으로만 가능하며, ϵ -TeX을 요구한다.

제 1 절 kotexplain.tex

kotexplain.tex은 플레인텍에서 한글을 사용하기 위한 매크로이다. 플레인텍은 라텍에 비하여 매우 가볍고 단순한 매크로이지만, 문서나 책을 만드는 데 쓰이기도 한다. 그 대표적인 예가 바로 Knuth 교수의 대표적인 시리즈 저서 *The Art of Computer Programming*과 *The TeXbook* (Knuth, 1984)이다.

플레인텍으로 작성된 텍 파일은 단순히 'tex foo.tex'과 같이 컴파일한다. 하지만 한글 사용을 위한 매크로 kotexplain.tex은 ϵ -TeX 엔진의 원시명령어(primitives)들을 이용해서 작성되었기 때문에 한글 플레인텍에 한해서는 tex 명령어가 아닌 etex 또는 pdftex 명령을 이용하여야 한다. 한글 플레인텍 문서를 작성하기 위해서는 작성하고자 하는 문서의 첫부분을

```
\input kotexplain
```

으로 시작하는 것만 빼면 보통의 플레인텍 문서를 만드는 것과 완전히 동일하다. 플레인텍의 사용법은 그 유명한 *The TeXbook*을 참고한다.

한글 매크로 kotexplain.tex을 이용하여 한글 글꼴을 설정하는 방법은 원래의 플레인텍에서 \font 명령을 이용하는 그것과는 좀 다르다. 예를 들어 알아보자. 한글 글꼴은 다음의 세 가지 방법 중 하나로 정할 수 있다.

- \hfont{outbtb}{at 12pt} 이 명령어는 현재의 한글 글꼴을 굵은(bold) 글꼴로 하고, 그 크기는 12포인트로 변경하라는 명령이며, 세 가지 방법 중 가장 일반적인 방법이다.
- \hfontname{outbtb} 이 명령어는 현재 글꼴의 크기를 그대로 유지한 체로 글꼴 모양만 굵을 글꼴로 변경하라는 의미이며, 이는 \hfont{outbtb}{}와 동일하다.

- `\hfontsize{at 12pt}` 이 명령은 현재 글꼴의 모양을 그대로 유지하면서, 크기만 현재의 크기에서 12 포인트로 변경하라는 뜻이다. `\hfont{}{at 12pt}` 와 동일하다.

제 2 절 hangulcweb.tex

플레인텍으로도 멋진 문서를 만들 수 있으나, 플레인텍으로 문서나 책을 만드는 일은 매우 수고로운 작업이 될 수 있기 때문에 대부분의 경우는 쉽게 사용할 수 있는 라텍을 이용한다. 하지만 반드시 플레인텍을 사용하여야 하는 경우가 있다. 예를 들어, 문학적 프로그래밍의 대표적인 시스템 CWEB은 플레인텍 매크로들로 이루어져 있기 때문에 CWEB 프로그래밍을 하려면, 플레인텍을 사용하여야 한다. 또 GNU의 여러 프로그램들의 사용 설명서를 만드는 표준인 *texinfo* 또한 플레인텍을 이용하여야 한다.

여기서는 한글 CWEB 프로그래밍을 하는 방법을 간단히 알아보자. 한글 플레인텍에서와 마찬가지로 작성하고자 하는 웹파일의 처음을

```
\input hangulcweb
```

으로 시작하면 된다. 사실은 `hangulcweb.tex`이 `kotexplain.tex`을 포함하고 있기 때문에 한글 CWEB 프로그래밍이 가능한 것이다. 한글 CWEB 프로그램의 매크로 파일 `hangulcweb.tex`은 CWEB 프로그램의 여러 영문 메시지를 한글화하였고, 가장 큰 특징은 CWEB 프로그램이 갖는 특징 중 하나인 PDF 기능을 한글도 자유롭게 이용할 수 있도록 구현해 놓았다는 점이다.

제 11 장 html 제작

tex 원본 문서로부터 XML 또는 HTML을 제작하는 방법은 몇 가지가 알려져 있다. 그 가운데, Eitan Gurari 씨가 만든 TeX4ht¹는 T_EXLive에 포함되어 있는 사실상 표준 도구이다.

현재 버전의 ko_TE_X/utf는 TeX4ht를 부분적으로 지원한다.² 즉, ko_TE_X의 한글 폰트들을 TeX4ht에서 처리할 수 있도록 하는 부수 파일을 제공하고 있다. ko_TE_X을 사용하여 작성한 문서를 HTML로 변환하려면 다음과 같은 명령을 사용하면 된다.

```
$ htlatex foo.tex "dhucs,html4" " -cunihtf"
```

이 명령을 이용하면 수백 개의 .htf 폰트 없이도 한글 HTML을 만들어낼 수 있다.³

TeX4ht에서 의도대로 원하는 HTML을 얻으려면 상당한 설정을 추가적으로 행해야 한다. 즉, 위의 해결책은 HTML/XML로 변환하는 모든 문제에 대한 해결책을 제공하는 것이 아니다. 그런 문제들은 ko_TE_X 자체와 관련된 것이 아니라 TeX4ht 설정과 더 깊이 관련되어 있을 것이기 때문이다. 표준 라텍 클래스의 문서라면 위에 제시한 정도로 큰 문제없이 HTML을 얻을 수 있겠지만 복잡한 문서의 경우 의도대로 출력을 얻기 위해 추가적인 설정을 해야 할 때가 있을 것이다. 자신의 경험을 다른 사람과 공유한다면 해결책을 더 빨리 찾아나갈 수 있을 것으로 생각한다.

¹<http://www.cse.ohio-state.edu/~gurari/TeX4ht/mn.html>

²<http://www.ktug.or.kr/jsboard/read.php?table=contrib&no=4313>

³이전에는 한글을 위한 TeX4ht용 글꼴을 만들어서 제공하는 방식을 썼던 적도 있다.

참고 문헌

- Knuth, D. E. (1984). *The T_EXbook*. Reading, MA, USA: Addison-Wesley.
- Knuth, D. E. (1986). *The T_EXbook* (Vol. A). Reading, MA, USA: Addison-Wesley.
- Knuth, D. E. (1986). *T_EX: The program* (Vol. B). Reading, MA, USA: Addison-Wesley.
- Knuth, D. E. (1986). *The METAFONTbook* (Vol. C). Reading, MA, USA: Addison-Wesley.
- Knuth, D. E. (1986). *METAFONT: The program* (Vol. D). Reading, MA, USA: Addison-Wesley.
- Kopka, H. (1991). *L^AT_EX—eine einföhrung* (Third ed.). Bonn, Germany: Addison-Wesley Verlag.
- Lamport, L. (1985). *L^AT_EX—A document preparation system—user’s guide and reference manual*. Reading, MA, USA: Addison-Wesley.
- Mittelbach, F., Goossens, M., Braams, J., Carlisle, D., Rowley, C., Detig, C., et al. (2004). *The L^AT_EX companion* (Second ed.). Reading, MA, USA: Addison-Wesley.
- Scherber, P. (1994). *Offizin—schriftenreihe zu T_EX, L^AT_EX, und metafont (band 1, hrsg. von DANTE e.V)*. Reading, MA, USA: Addison-Wesley.
- Thành, H. T. (2000). *Micro-typographic extensions to the T_EX typesetting system*. Unpublished doctoral dissertation, Masaryk University.
- Thành, H. T. (2001). Margin kerning and font expansion with pdfT_EX. *TUGboat*, 22(3).
- Wilson, P. (2006). *The memoir manual*.
- 김강수. (2004). *hangul-k* 사용자 안내서.
- 김강수. (2007a). L^AT_EX에서 한글 유니코드 문서 작성하기.
- 김강수. (2007b). 한글 문장부호의 조판 관행에 대하여. *The Asian Journal of T_EX*, 1(1), 17–30.
- 김강수, & 주철. (2004:2007). 한글 L^AT_EX에서 트루타입 폰트 사용하기.
- 김도현. (2004). Omega(Ω)를 이용한 한글 문서 조판 — dhangul을 중심으로. *KTUG 뉴스 레터*, 1(1), 2–15.
- 김도현. (2007). 폰트의 저작권이 T_EX 출력물에 미치는 영향. *The Asian Journal of T_EX*, 1(1), 45–56.

- 박지원. (1780). 하룻밤에 아홉 번 물을 건너다.
- 은광희. (2005). 한글 라텍 길잡이.
- 이기황. (2007). Hangul-ucs 를 이용한 옛한글 조판. *The Asian Journal of T_EX*, 1(1), 31-43.
- 이종운. (1991). 도서편집총람 (1st ed.). 범우사, 서울.
- 최우형. (1992). *hlatex* 가이드.
- 최현배. (1999). 우리말본 (18th ed.). 정음문화사.
- 한글 맞춤법. (1988). 문교부 고시 제 88-1 호.
- 한글연구회. (1992). 새 한글 맞춤법 및 옹례집 (1st ed.). 이사야 법조각.
- 홍윤표, 이기문, 김충희, 박성종, 정진상, 김병선, et al. (1991, 8). 한글 옛글자의 컴 퓨터 처리방안에 대한 연구. 한국어전산학, 창간호.

찾아보기

【 F 】	방법	44
french-spacing	35	검색과 추출
		35
【 H 】		고아 회피
halphabst	86	58
hbjbtex	106	글꼴
HFSS	77	CM
\hindexhead	111	34
		EC 글꼴
【 K 】		34
\ksnamedef	88	pxfonts
		35
【 L 】		T1 인코딩
LaTeX3 개발	10	34
		영문 글꼴
【 M 】		34
\MapHangulFamily	80	은 글꼴
		40
【 N 】		글자체
NFSS2	75	구성 요소
nonfrench spacing	47	75
		기본 글꼴 설정을 변경
【 O 】		38
overfull	56	기본적으로 사용되는 글꼴
		36
【 P 】		【 ㄴ 】
pdf bookmark	60	내형 글자체 바꿈 명령
		77
【 S 】		【 ㄷ 】
softbold	80	다국어 문서
strictcharcheck	41	63
		단어 간격
【 T 】		97
TDS	17	단어간격
		36, 43
【 ㄱ 】		【 ㄹ 】
각주 문단	95	라텍
각주 판짜기	48, 84	10
강조	43	개발 목표
		10
글꼴 변경	44	글자체 선택
기울인 글꼴	43	75
밑줄	45	버전 3
		10
		【 ㅁ 】
		매개 모음
		85
		명령행
		34
		명령형
		40, 76
		문서 식자 체계
		텍
		9
		문서의 모양새
		72
		문장 부호의 타이포그래피
		54
		문헌 인용
		105
		문헌목록
		60
		미세 타이포그래피
		52

finemath	52	【 ㅈ 】	
마이크로 타이포그래피	54	자간	36, 46, 96
마진 커닝	54	자동 음운 변화	85
수식 간격	52	색인	87
폰트 확장	54	자동 음운 변화 처리	86
【 ㅊ 】		자동 조사	
방침	44	인용	86
【 ㅋ 】		자동조사	49
색인	60, 111	장과 절 포맷	35
\hindexhead	111	장절 명령	43
서식	43	장절명령	43
선언형	76	조사 이형태	85
설치	16	조판	47
【 ㆁ 】		줄간격	74
엄격한 문자 체크	41	중괄호	40
여는 괄호	56	중국어	63
역사		지정사 탈락	85
\TeX	10	짧은 간격	64
$\TeX 2_{\epsilon}$	10	【 ㆏ 】	
\TeX	9	참고문헌	61
hlatex	10	추가 선택 처리 방식	72
한글 라텍	11	【 ㆑ 】	
옛한글	61	카운터 형식	48
Ω	12	ko \TeX	
외형 가족	80	HTML 제작	118
외형 글자체 바꿈 명령	77	plain \TeX	116
우리말 글자체 가족	79	ko \TeX /utf	
우리말 이름	88	옵션	35
유니코드	33	ko.TeX/euc	
UTF-8	33	사용법	71
은 글꼴 이외의 글꼴	36	클래스	
이탤릭 글자	43	amsart	62
fake slanted	44	amsbook	62
인명		memoir	66
Knuth	116	oblivoir	37, 38, 66
김도현	33	【 ㆒ 】	
남수진	116	타이포그래피	46
신정식	56	통합 한글 텍 패키지	14
이기황	134	【 ㆓ 】	
일본어	63	파일	
일본어 문단	63		

.dvi.....	34	kotexplain.....	17
.log.....	35, 41	makeidx.....	111
.tfm.....	35, 40, 41	memhangul-ucs.....	66
dhucs-sectsty.....	43	microtype.....	55
firsttest.tex.....	33	myulem.....	99
hyperref.cfg.....	58, 63	natbib.....	61
komkindex.pl.....	60	newprosper.....	63
obatang.ttf.....	61	paralist.....	36, 51
ogulim.ttf.....	61	powerdot.....	63
prosper.cls.....	63	prosper.....	63
패키지		ruby.....	63
amsmath.....	62	sectsty.....	43, 100
apacite.....	61	setspace.....	74, 98
babel.....	41	showkeys.....	91
beamer.....	62, 63	ulem.....	45, 99
bigfoot.....	95	폰트의 남용.....	39
cite.....	61		
CJK.....	63, 64	【 ㅎ 】	
dhucs.....	33	한글	
dhucs-cmap.....	59	완성형.....	124
dhucs-enumerate.....	36	한글 PDF 책갈피.....	33
dhucs-gremph.....	37, 44	한글 라텍스.....	11
dhucs-interword.....	33, 46, 96-98	글자체 바꿈 명령.....	81
dhucs-midkor.....	61	글자체 선택.....	74
dhucs-paralist.....	51	글자체 종류.....	79
dhucs-sectsty.....	100	단원 판짜기.....	88
dhucs-setspace.....	46, 95, 98	문서의 틀.....	71
dhucs-trivcj.....	63	분절.....	89
dhucs-ucshyper.....	60	숫자 모듬 명령.....	83
dhucsf.....	48	추가 선택.....	71
enumerate.....	50, 51, 84	한글 이름.....	43
enumitem.....	36, 51	행간.....	36, 43, 46
eurofont.....	41	행나눔.....	56
eurosym.....	41	환경형.....	76
fapapersize.....	68		
footmisc.....	95		
hfontsel.....	38		
hlatex-interword.....	97		
hsectsty.....	100		
hsetspace.....	74, 95, 98		
hyperref.....	33, 50, 58, 60, 63, 74, 90		
ifpdf.....	33		
kotex.....	33		

[illegible][illegible][illegible]

[부록] Windows 유니코드 편집기

TeX 작업을 가장 편리하게 할 수 있는 편집기 중의 하나로 단연 UN*X 세계에서는 Emacs와 AUCTeX을 들 것이다. 그러나 윈도우즈 운영체제에서도 Emacs를 사용할 수는 있지만 지배적 위치를 점하고 있지는 못하다.⁴

koTeX 사용에 있어서 편집기는 반드시 유니코드/UTF-8 인코딩으로 파일을 저장하고 불러오고 편집할 수 있어야 한다. TeX 편집을 위한 기본 기능이 지원되어야 좋을 것은 말할 나위도 없을 것이다.

가장 많은 사용자를 가지고 있는 윈도우즈 라텍 편집기는 WinEdt⁵이다. 비록 자유 소프트웨어는 아니지만 저렴한 가격에 비하여 성능이 상당히 우수하고 정말 편리한 라텍 작업 환경을 제공하는 라텍에 특화된 일종의 통합 개발 환경이다. 오래도록 이 편집기가 유니코드 입출력이 어려워서 상당히 곤란을 겪었으나 최근 버전은 (제한적이기는 하나) 유니코드/UTF-8 파일을 저장하고 편집할 수 있다. 제한적이라 한 것은, 윈도우즈에서 현재 활성화되어 있는 코드 페이지(한글 윈도우즈의 경우 CP949)에 해당하는 범위의 문자만을 유니코드/UTF-8로 저장할 수 있다는 점인데, 이 때문에 일본식 한자 등이 섞여 있는 문서는 문제가 생길 수 있다. 그렇지만 일반적인 한글 문서는 잘 처리한다. 문서 시작 부분에서

```
% -*- TeX:UTF-8 -*-
```

이라는 문서 포맷 표지를 붙이고, 저장할 때 UTF-8 인코딩을 잘 선택하면 큰 무리없이 사용할 수 있다.

KC2006에서 기본 편집기로 상정하고 있는 EmEditor⁶도 가격 대 성능비가 대단히 훌륭하다.⁷ 이 편집기는 TeX 관련 기능을 plug-in으로 제공하고 있으며, TeX 전용 편집기는 아니지만, MTeXHelper2라는 plug-in을 조금만 손보면 아주 잘 사용할 수 있다. EmEditor는 처음부터 유니코드 편집기를 표방하고 있으므로 WinEdt과 같은 제한이 없이 유니코드 문서를 잘 작성하고 처리할 수 있다.

최근 주목받고 있는 LyX⁸이라는 프로그램도 일별할 가치가 있다. 이 프로그램은 원래 Linux 세계에서 TeX Front End WYSIWYG⁹ 에디터로 유명했던 것인데, 버

⁴그럼에도 불구하고, Windows에서도 Emacs는 여전히 가장 편리한 TeX 작업 환경을 제공해준다. 유니코드의 입출력에도 큰 문제는 없다. 다만, 완성형 문자 범위를 넘는 한글을 입력하는 것이 어렵다는 점만 제외한다면.

⁵<http://www.winedt.com>

⁶<http://www.emeditor.com>

⁷무료로 사용할 수 있는 버전도 있으나 빠진 기능이 많다.

⁸<http://www.lyx.org>

⁹LyX의 용어로는 WYSIWYG이 아니라 WYSIWYM (What You See Is What You Mean)이라 한다.

전 1.5.0부터 유니코드를 지원함으로써 $\text{ko}\text{T}_{\text{E}}\text{X}$ 을 위한 에디터로 손색없이 쓸 수 있다. 특히 이 프로그램의 장점은 라텍 명령과 텍스트를 함께 써넣는 문서 작성과는 조금 다르게 메뉴와 버튼을 이용해서 대부분의 라텍 관련 콘트롤 시퀀스를 생성해낸다는 점이다. 라텍에 익숙하지 않은 분도 워드 프로세서처럼 쓸 수 있다. 특히, LyX은 오픈소스 프로그램이라는 것이다! 이와 유사한 것으로 상업용 프로그램이고 가격이 비교적 높은 편인 Scientific Word와 같은 것이 있다.

이밖에도 좋은 에디터가 많이 있을 것이다. 에디터에 익숙해지는 것은 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 을 잘 사용하기 위한 출발점이다.

[부록] 이 문서에 대하여

ko_{TeX} 사용설명서는 은광희의 “한글 라텍 길잡이”와 김강수의 “ \LaTeX 에서 한글 유니코드 문서 작성하기”를 합친 것이다. 전자는 한글 라텍 (H \LaTeX)의 사용설명서였고 후자는 Hangul-ucs의 패키지 문서였다. 두 문서의 장점을 취합하려 노력하였으며, 가능한 원래 텍스트의 모습을 유지하려 하였다. 이 문서는 현재 공저자 중 김강수에 의해 관리되고 있다.

이 문서는 utf 버전과 euc 버전으로 컴파일할 수 있다. 다만, euc 버전으로 컴파일했을 때는 utf 부분 사용 설명서가 모두 생략된다. euc 버전은 단지 테스트를 위한 것이므로, 최종본을 얻으려면 utf 버전으로 컴파일하기 바란다.

euc 버전으로 컴파일하려면 현재 작업 디렉터리에 `definekg.eucmode`라는 파일을 만들면 된다. 내용은 상관이 없다.¹⁰

```
$ echo "no contents" > definekg.eucmode
```

이 파일을 삭제하면 utf 모드로 컴파일한다.

pdf 제작은 DVIPDFMx 또는 pdf \LaTeX , 어느 쪽이든 가능하다. 다만 DVIPDFMx를 이용하려 한다면 `fig/` 디렉터리 안의 그림들에 대하여 미리 `.bb` 파일을 만들어두는 절차를 거쳐야 한다. `ebb`, `xbb` 등의 유틸리티를 이용하면 된다. 만약 `xbb`가 확장명 `.xbb`인 파일을 만들어낸다면 `kotexguidebody-*.tex`의 제25행부터 정의되어 있는 `\DeclareGraphicsRule` 명령의 bb파일 확장명을 `.xbb`로 수정한다. pdf \LaTeX 으로 컴파일한 pdf 문서는 그 크기가 좀 커질 것이다.

이 문서는 ko_{TeX} 프로젝트의 일부로서, LPPL version 3 또는 그 이후 버전의 라이선스로 배포된다.

이 문서의 내용에 대한 코멘트는 KTUG¹¹ 게시판에 글을 올리거나 저자¹²에게 메일을 보내주기 바란다.

¹⁰배포되는 소스 파일에서 EUC-KR 파일들을 생성해내는 `generateeucfiles.sh`를 실행하라.

¹¹<http://www.ktug.or.kr>

¹²<mailto:info@mail.ktug.or.kr>

[부록] 글꼴 예문

기본 글꼴

은 바탕

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃
눈시린 유리창마다/ 톱밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기
에 쿨럭이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 톱밥을 불빛 속에 던져 주
었다.

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃
눈시린 유리창마다/ 톱밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기
에 쿨럭이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 톱밥을 불빛 속에 던져 주
었다.

學而時習之, 不亦說乎.

學而時習之, 不亦說乎.

은 돋움

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃
눈시린 유리창마다/ 톱밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기
에 쿨럭이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 톱밥을 불빛 속에 던져 주
었다.

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃
눈시린 유리창마다/ 톱밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기
에 쿨럭이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 톱밥을 불빛 속에 던져 주
었다.

學而時習之, 不亦說乎.

學而時習之, 不亦說乎.

은 타자

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수
꽃 눈시린 유리창마다/ 톱밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은
감기에 쿨럭이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 톱밥을 불빛 속에
던져 주었다.

은 그래픽

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃

눈시린 유리창마다 / 툇밥 난로가 지퍼지고 있었다. / 그믐처럼 멎은 졸고 / 멎은 감기에 쿨럭이고 / 그리웠던 순간들을 생각하며 나는 / 한 줌의 툇밥을 불빛 속에 던져 주었다.

막차는 좀처럼 오지 않았다. / 대합실 밖에는 밤새 송이눈이 쌓이고 / 흰 보라 수수꽃 눈시린 유리창마다 / 툇밥 난로가 지퍼지고 있었다. / 그믐처럼 멎은 졸고 / 멎은 감기에 쿨럭이고 / 그리웠던 순간들을 생각하며 나는 / 한 줌의 툇밥을 불빛 속에 던져 주었다.

추가 글꼴

은 불

막차는 좀처럼 오지 않았다. / 대합실 밖에는 밤새 송이눈이 쌓이고 / 흰 보라 수수꽃 눈시린 유리창마다 / 툇밥 난로가 지퍼지고 있었다. / 그믐처럼 멎은 졸고 / 멎은 감기에 쿨럭이고 / 그리웠던 순간들을 생각하며 나는 / 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 옛글

막차는 좀처럼 오지 않았다. / 대합실 밖에는 밤새 송이눈이 쌓이고 / 흰 보라 수수꽃 눈시린 유리창마다 / 툇밥 난로가 지퍼지고 있었다. / 그믐처럼 멎은 졸고 / 멎은 감기에 쿨럭이고 / 그리웠던 순간들을 생각하며 나는 / 한 줌의 툇밥을 불빛 속에 던져 주었다.

學而時習之，不亦說乎.

은 궁서

막차는 좀처럼 오지 않았다. / 대합실 밖에는 밤새 송이눈이 쌓이고 / 흰 보라 수수꽃 눈시린 유리창마다 / 툇밥 난로가 지퍼지고 있었다. / 그믐처럼 멎은 졸고 / 멎은 감기에 쿨럭이고 / 그리웠던 순간들을 생각하며 나는 / 한 줌의 툇밥을 불빛 속에 던져 주었다.

學而時習之，不亦說乎.

은 신문

막차는 좀처럼 오지 않았다. / 대합실 밖에는 밤새 송이눈이 쌓이고 / 흰 보라 수수꽃 눈시린 유리창마다 / 툇밥 난로가 지퍼지고 있었다. / 그믐처럼 멎은 졸고 / 멎은 감기에 쿨럭이고 / 그리웠던 순간들을 생각하며 나는 / 한 줌의 툇밥을 불빛 속에 던져 주었다.

學而時習之, 不亦說乎.

은 자모바탕

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 자모돋움

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 자모소라

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 자모노벨

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 필기

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 펜글씨

막차는 좀처럼 오지 않았다./ 대합실 밖에는 밤새 송이눈이 쌓이고/ 흰 보라 수수꽃 눈시린 유리창마다/ 툇밥 난로가 지퍼지고 있었다./ 그믐처럼 멎은 줄고/ 멎은 감기에 쿨적이고/ 그리웠던 순간들을 생각하며 나는/ 한 줌의 툇밥을 불빛 속에 던져 주었다.

은 펜홀림

막차는 증처럼 오지 않았다./ 대합실 밖에는 밤새 층이늘이 쌓이고/ 흰 보라 수수꽃
늘어진 우리창가다/ 뚝뚝 난로가 끼적지고 있었다./ 그뚝처럼 맺은 줄고/ 맺은 감기
에 클럭이고/ 그리웠던 순간들을 생각하여 나는/ 한 줌의 뚝뚝을 불빛 속에 던져 주
었다.

[부록] 용어 대조

이 글의 배경을 이루는 두 문서는 같은 단어에 대하여 서로 다른 용어나 역어를 쓴 것이 있다. 원본 문서의 형태를 최대한 유지한다는 원칙으로 이 문서가 작성된 까닭에 이러한 용어의 불일치를 소거하지 않고 문서 중 모두 노출되도록 하였다(그 중 일부는 처음 출현할 때 영어 단어를 병기함으로써 혼란을 줄이도록 하였다). 그러나 이로부터 야기될 독자의 혼선을 피하자는 뜻에서 몇 가지 용어의 대조표를 제시한다.¹³ 문서 A와 문서 B라 함은 반드시 특정 문서를 지칭하는 것은 아니다.

용어 대조		
용어	문서 A의 용어	문서 B의 용어
bookmark	북마크, 책갈피	책갈피
cite key	참조자	참고 문헌 표지
code system	코드 체계	부호 체계
default	디폴트, 기본(값)	애초(값)
document class	문서 클래스	문서 종류
driver	드라이버	구동기
embed	임베드, 내장	내장
encoding	인코딩	부호화
encoding system	인코딩 체계	부호 체계
external	—	외형
font	폰트, 글꼴	글자체
font encoding	폰트 인코딩	글자체 부호화
font family	글꼴 가족	글자체 가족
font selection scheme	폰트 선택 스킴	글자체 선택 방식
hyphen	하이픈	붙임표
implementation	구현	구현
inter-character space	자간	자간
inter-word space	단어간격, 어간	어간
internal	—	내형
label	레이블	이름표
layout	레이아웃	틀잡기
lead	행간	줄간격

¹³용어 선정에 도움을 준 이기황 박사께 감사한다.

용어 대조 (계속)

용어	문서 A의 용어	문서 B의 용어
line breaking	행나눔, 행자름, 개행	—
macro	매크로	모듬 명령
micro-typography	미세 타이포그래피	—
option	옵션	추가선택
package	패키지	꾸러미
patch	패치	깁기
page	페이지	쪽
preamble	프리앰블	전문 영역
preprocessor	전처리기	앞처리기
source	소스, 원본	원천
source code	소스 코드	원천 코드
style	스타일, 형식	모양새
symbol (character)	기호 (문자)	상징 기호
system	시스템, 체계	체계
truetype	트루타입	—
typesetting	조판 (식자)	판짜기
typesetting system	조판 시스템	식자 체계
typography	타이포그래피	—
ℒ _{TeX}	레이텍	라텍
PUA (private use area)	PUA, 사용자 영역	—
Unicode	유니코드	유니코드
T _E X implementation	텍 배포판	텍 구현
—	자동 조사	자동 음운 변화 처리
—	서술격 조사 어간	지정사