

한글 \LaTeX 에서 임의의 트루타입 한글 폰트 사용하기

김강수 주철

2007년 7월 6일

일러두기

이 글은 2003년에 썼던 “ \LaTeX 에서 임의의 트루타입 글꼴 사용하기”라는, 김강수·주철 공저의 글을 고쳐쓴 것이다.

한글 트루타입 폰트를 \LaTeX 에서 사용할 수 있게 된 과정은 이전의 글에 정리되어 있다. 이 글은 2003년 이후 달라진 한글 \LaTeX 환경을 위한 것이므로 해당 사항에 대해서는 이전 글을 참조하기 바란다.

2007년 6월까지, `hangul-ucs`에서 완성형 인코딩 폰트를 사용하지 못하는 것으로 알려져 있었다. 그러나 이 글은 KTS 제1차 워크샵의 결과로서 이 일을 가능하게 한 성과가 반영되어 있다.

차 례

차 례	1
1 들어가는 말	2
1.1 제한	2
1.2 많은 글꼴?	3
2 준비	3
2.1 용어	4
2.2 글꼴의 대응	5
2.3 작업 준비	6
3 절차	7
3.1 \TeX 폰트 이름	7
3.2 TFM 텍 폰트 파일 얻기	7
3.3 FD 폰트 정의 파일 만들기	9
3.4 map 파일의 작성	10
3.5 테스트	12
4 <code>ttf2kotexfont</code> script	13
5 마치는 말	15

1 들어가는 말

KTUG¹⁾이 활동을 시작한 이후, 한글 T_EX, L^AT_EX 사용 환경은 몰라보게 달라졌다. 주요한 점들만 꼽아보아도,

- 플랫폼에 관계없이 쉽고 간편하게 T_EX 시스템을 설치하고 운영할 수 있게 되었다. KC2006을 통하여 한글을 운영하는 T_EX 시스템의 설치와 관리 및 운영을 놀랍도록 편리하게 개선하였다.
- 한글 트루타입 폰트를 자유롭게 사용할 수 있게 됨으로써 폰트의 제약을 크게 극복하게 되었다.
- 유니코드를 한글 식자에 이용하게 됨으로써 모든 한글과 옛한글 문자 및 다국어 조판도 가능하게 되었다.
- 이밖에 종래 불가능하다고 알려져 있던 거의 대부분의 문제가 해결되어 왔다.

이 글은 그 가운데 트루타입 폰트를 한글L^AT_EX 패키지와 함께 사용하는 방법에 관한 간단한 사용자 안내서 (User's Guide)이다.

1.1 제한

현재 L^AT_EX 작업환경에서 한글을 사용할 수 있도록 하는 방법은 여러 가지가 있다. 그 방법을 모두 다룰 수는 없으므로, 이 글은 다음과 같은 범위에서만 필요한 정보를 제공하려 한다.

- ① 한글L^AT_EX 이라 함은 Hangul-ucs 패키지를 의미한다.²⁾
- ② 입력 인코딩은 유니코드 UTF-8 인코딩된 한글 문서만을 다룬다.
- ③ 최종 출력은 PDF로 한다. PDF를 얻는 방법은 DVIPDFM_x를 이용하는 방법을 기본으로 하되, pdfL^AT_EX에 의한 방법도 고려하기로 한다.
- ④ ttf 글꼴들은 대부분 저작권이 있고, 자유롭게 공개된 ttf는 그 수가 매우 적다. 이 글은 글꼴의 저작권에 대한 문제는 다루지 않는다. 그러므로 저작권과 관련된 문제는 사용자 자신이 책임을 져야 할 것이다.

1) 한글 T_EX 사용자 그룹. Korean TeX Users Group. <http://www.ktug.or.kr>.

2) 이 문서의 2003년 버전은 H^AL^AT_EX을 전제로 하였다. 그 당시는 hangul-ucs가 개발되기 전이었고, 따라서 한글 문서의 기본 인코딩도 euc-kr이었다.

이 글은 실제로 사용자가 “따라하기” 위한 지침은 아니다. 실용적으로 트루타입을 사용하려면 김도현 교수가 작성한 `ttf2kotexfont` 유틸리티로 충분하다. 그러나, 이 스크립트가 어떤 작용을 하는지에 대하여 궁금해하는 분들이 있어³⁾ 그 과정을 설명하기 위하여 작성하는 것이다. `ttf2kotexfont` 스크립트 사용법은 제4 절에 설명해두었다.

1.2 많은 글꼴?

사용자들의 일반적인 요구는 아마도 “워드에서 쓰듯이 글꼴을 자유롭게 사용” 하는 방법을 적용해보고 싶어하는 것인 듯하다. 그러나 이 문제는 한번 생각해봐야 한다. 과연 얼마나 많은 글꼴이 필요한 것인가?

L^AT_EX은 포스터나 광고물, 또는 초대장이나 생일카드를 만드는 데는 적합하지 않은 것으로 알려져 있다. 물론 불가능하지는 않겠지만, 여러 종류의 글꼴이나 또는 적은 수라도 특별하게 디자인된 글꼴을 사용해야 하는 상황은 아무래도 L^AT_EX 작업과는 거리가 멀다는 느낌이 든다. 여기서 문제삼는 문서는 주로 상호참조가 필요한 과학적 문헌이거나 큰 분량의 전문서적일 것이다. 이런 곳에서 사용되는 글꼴은 두세 종류면 충분할 것이다.

표준적인 L^AT_EX에서 사용되는 글꼴은 `\rm`, `\sf`, `\tt` 이 세 종류뿐이다. 강조를 위해서 기울임(이텔릭) 처리가 추가되거나 **boldface**, SMALL CAPITAL을 필요하다면 쓰는 정도. 다만 `\rm`에 어떤 폰트를 적용할 것인가 하는 문제는 디자인 문제라고 생각한다. 평범한 Computer Modern 글꼴을 쓸 수도 있고, Times 글꼴이나 Palatino 글꼴을 적용할 수도 있는 것이다.

한글로 만들어진 문서의 경우에도 크게 다를 것은 없다. 기본적으로 바탕글(본문)을 위한 글꼴, 강조를 위한 글꼴, 인용을 위한 글꼴 이 세 가지 정도가 필요할 것으로 생각한다. 글꼴 사용을 결정할 때는, 이 세 가지에 각각 적절한 폰트를 할당하는 형태로 접근하는 것이 옳다.

한글 원도가 바탕, 돋움, 굴림, 궁서라는 네 가지 글꼴을 기본으로 제공하는 것을 고려하여, 여기서는 이와 유사하게 네 가지 글꼴을 문서에서 사용하는 방법을 알아보려 한다. 그 각각을 바탕(또는 명조), 돋움(또는 고딕), 굴림, 장식 글꼴이라고 부르기로 한다.

그러나, 어떤 경우든 한 문서에 지나치게 많은 글꼴이 사용되면, 오히려 문서의 품위를 떨어뜨린다는 것을 잊지 말자.

2 준비

이 작업을 따라하기 위해서는 먼저 다음과 같은 것이 준비되어 있어야 한다.

3) Kunggom 님. 실제로 이 글을 따라해보고 초고의 문제점을 지적해주셨다.

표 1: sfd와 폰트 인코딩의 대응

문자집합	폰트 인코딩	sfd
EUC-KR	EUC-KR	KS-HLaTeX.sfd
EUC-KR	Unicode	UKS-HLaTeX.sfd
Unicode BMP	EUC-KR	UKS-UCS2.sfd
Unicode BMP	Unicode	UCS2.sfd

TeX 시스템 윈도 플랫폼의 경우, KC2006을 권장한다. 다만 MiKTeX은 호환성의 문제로 인하여 이 글에서 설명하는 모든 결과가 다 잘 된다고 보장할 수 없다는 점을 적어둔다. ttf2tfm이나 DVIPDFMx는 TeX 시스템의 일부로서 이미 설치되어 있을 것이다.

한글LaTeX Hangeul-ucs가 설치되어 있어야 한다. 4.0.0 이후 버전을 권장한다. KC2006에는 한글LaTeX이 이미 잘 설치·설정되어 있다.

폰트 파일들 Truetype 한글 글꼴이 필요하다. 어떤 글꼴을 필요로 하느냐는 다음에 다시 언급한다.

2.1 용어

이 글에서 사용되는 약자와 용어 등에 대하여 간단히 적어둔다.

%HOME% 사용자의 홈 디렉토리이다. 리눅스 또는 맥에서는 \$HOME으로 표기한다. Windows KC2006의 경우, 별도로 환경변수를 지정하지 않으면 [TeX]\HOME이 자동으로 설정되어 쓰인다.

\$ 명령행 프롬프트를 의미한다. Windows에서는 cmd 창을 열고 명령을 직접 써넣는다는 의미이다.

시스템 폰트 폴더 Windows의 경우 대부분 C:\Windows\Fonts이다. 이 글꼴 폴더에 있는 글꼴은 KC2006이 잘 인식한다.

입력 인코딩(input encoding) TeX으로 하여금 처리하게 할 파일에 입력된 문자의 인코딩을 말한다. 여기서는 Unicode/UTF-8을 의미한다. 이 인코딩의 문자 부호를 저장하거나 편집할 수 있게 하는 편집기가 필요하다.

폰트 인코딩(font encoding) 폰트 파일 안에 문자 집합이 어떤 인코딩을 따라서 배열되어 있는가를 나타내는 것이다. 이 글에서 문제삼는 한글 폰트는 유니코드 순서대로 배열된 유니코드 인코딩 폰트와 EUC-KR 코드에 따라 배열된 “완성형 인코딩 폰트” 두 종류이다.

표 2: 글꼴 대응

글꼴	보통 서체	두꺼운 서체
바탕글꼴	윈도바탕	한양태명조
돋움글꼴	윈도돋움	휴먼태고딕
굴림글꼴	윈도굴림	한양태동근고딕
장식글꼴	양재난초M	

서브폰트 한글 문자 집합은 상당히 많은 글자들로 이루어져 있다. 이를 \TeX 이 다룰 수 있게 하기 위해 각각 256 문자씩 잘라내어서 취급하는 방법을 말한다.

서브폰트 정의 (sfd) 서브폰트로 잘라낸 각 부분에 어떤 문자가 대응하는가를 지시하는 정의 파일이다. \HATeX 에서는 EUC-KR 한글을 다루기 위해 기호문자 04번에서 07번, 한글 0번에서 9번, 한자 10번에서 29번으로 이루어진 sfd 정의를 사용하여 서브폰트를 취급하며, hangul-ucs에서는 유니코드 BMP 영역 문자를 00번에서 ff번까지 256 문자씩 차례로 잘라낸 서브폰트 정의를 이용한다. 표 1을 참고하라. EUC-KR 문자집합을 다루는 것이 \HATeX 이고 Unicode 한글을 다루는 것이 hangul-ucs이다.

2.2 글꼴의 대응

한글 폰트의 사용에 있어서 문제가 되는 것은, 조진환 교수가 지적한 바와 같이,⁴⁾ 볼드체(두꺼운 글꼴)가 없다는 점이다. \DVIPDFMx 에서 글자의 획을 늘려서 처리해줄 수(이른바 *fake-bold*) 있지만 *fake-bold*의 품위가 만족스럽지는 못하다.

그러므로, 볼드체 글꼴은 따로 설정해주는 것이 좋다. 예를 들면, 명조체에는 건명조체를 그 볼드체로 대응시키는 방법이다. 이 둘을 묶어서 하나의 글꼴 가족처럼 사용하자는 것이다.

이 글에서는 비교적 쉽게 구할 수 있는⁵⁾ 글꼴을 사용하여, 다음 표 2와 같이 설정해보기로 한다.⁶⁾

만약 굵은 글꼴을 대응시키기 어렵다면 차라리 굵은 글꼴 없이 쓰는 것이 나은 선택일 수도 있다. 왜냐하면 다른 글꼴을 대응시켰을 때 그 안에 포함된 문자가 완전히 동일하다는 보장도 없을뿐더러(특히 기호문자와 한자가 그러하다) 자칫 잘못하여 전혀 상관없는 글꼴을 대응시키게 되면 문서 품위에 문제가 생길 수도 있기 때문이다.

4) <http://www.ktug.or.kr/jsboard/read.php?table=setup&no=1005&page=1>

5) 글꼴을 어떻게 구할 수 있느냐 하는 문제는 이 글의 관심사가 아니다.

6) 볼드체 글꼴을 따로 설정하기 어렵다면, 기본 글꼴에서 \TeX 폰트 이름만 볼드체로 된 TFM 파일을 추출한 다음, \DVIPDFMx 에게 처리를 맡길 수 있다. 이에 관해서는 각주 18을 보라.

표 3: 폰트 파일 목록

폰트 이름	폰트 파일 이름
윈도바탕	MSBatang.ttf
한양태명조	h2mjrb.ttf
윈도돋움	MSDotum.ttf
휴먼태고딕	hmfbg.ttf
윈도굴림	MSGulim.ttf
한양태동근고딕	h2drrb.ttf
양재난초M	ynch05.ttf

윈도 돋움과 윈도 굴림은 하나의 ttc 파일⁷⁾에 포함되어 있으므로, 이 둘을 분리해야 할 필요가 있다.⁸⁾ 이를 위해서 breakttc.exe 라는 유틸리티가 필요하다.⁹⁾ breakttc.exe 를 gulim.ttc 에 대하여 실행하면 네 개의 파일이 분리되어 나오는데, 00에서 03까지 차례로, 굴림, 굴림체, 돋움, 돋움체에 해당한다. 그러므로 FONT00.ttf 를 MSGulim.ttf 로, FONT02.ttf 를 MSDotum.ttf 로 이름을 바꾸어서 사용한다. 윈도 바탕체는 batang.ttc 의 0번 폰트를 MSBatang.ttf 로 이름을 바꾸어 사용한다.

물론, 임의의 다른 ttf 를 사용하려 한다면, ttc 파일이 아닌 한, 그대로 사용할 수 있을 것이다.

2.3 작업 준비

적당한 위치에 새로운 디렉토리를 하나 만든다. 편의상 여기에서는 C:\makefont 라고 하기로 하자. 그리고 다음 순서를 따른다.

1. 필요한 ttf 파일들을 이 디렉토리로 복사한다. 위의 폰트 대응에 따를 때 필요한 파일의 목록은 표 3과 같다.
2. 트루타입을 이용하려면 sfd 파일이 필요하다. sfd 는 “sub-font definition” 의 약자로서, 한글 \LaTeX 이 많은 수의 한글 문자를 각각 256 개씩의 문자 영역(plane)

7) ttc 는 유사하거나 관련된 몇 개의 ttf 를 하나의 폰트 파일로 묶어 압축한 것이다. 각각의 폰트들은 index 번호로 참조된다. 한글 윈도에는 batang.ttc 와 gulim.ttc 라는 두 개의 ttc 폰트 파일이 기본으로 들어 있다. batang.ttc 는 바탕, 바탕체, 궁서, 궁서체, 네 개의 ttf 폰트가 묶여 있고, gulim.ttc 에는 굴림, 굴림체, 돋움, 돋움체, 네 개의 ttf 폰트가 묶여 있다.

8) 실제로는 반드시 분리하지 않더라도 ttf2tfm 의 -f 옵션을 쓸 수 있다. 이 옵션 다음에는 ttc 파일에 묶인 글꼴 인덱스를 지정하는데, batang.ttc 의 경우 바탕, 바탕체, 궁서, 궁서체가 각각 0, 1, 2, 3의 인덱스 값을 갖는다. 그러나 pdf \LaTeX 은 ttc 글꼴을 처리하지 못하므로, 이 글에서는 글꼴을 분리하여 사용하는 쪽을 선택하였다.

9) 도스용 프로그램인데, MS 에서 제공하는 TTSDK(ttstdk.zip)에 포함된 것. 리눅스에서는 ttc2ttf 라는 유틸리티가 있다. 다음 글을 참고하라. <http://mail.mizi.com/pipermail/devel/2002-January/004279.html>

표 4: T_EX 폰트 이름

폰트	보통	두꺼운	기울인
명조	omymjm	omymjb	omymjmo
고딕	omygtm	omygtb	omygmjo
굴림	omyglm	omyglb	omyglmo
장식	omygrm		

으로 조개어 다루기 위한 정의 파일이다. 우리의 작업을 위해서 필요한 sfd는 UCS2.sfd와 UKS-UCS2.sfd 두 종류인데, 이 가운데 UKS-UCS2.sfd는 비교적 최근에 만들어졌기 때문에 별도로 구해야 할 수 있으나 UCS2.sfd는 이미 시스템에 설치되어 있을 것이다.¹⁰⁾ UKS-UCS2.sfd는 2007년 한국텍학회 워크샵에서 논의되어 조진환 교수가 작성한 새로운 sfd이다. Hangul-ucs CVS에서 구할 수 있으며 KC2006은 kcupdate 형식으로 간단하게 설치할 수 있다.

3 절차

3.1 T_EX 폰트 이름

각 폰트 명칭에 해당하는 T_EX 폰트 이름을 정해주어야 한다. 이 폰트 명칭은 일정한 일관성이 있으면 좋은데, 아직까지 표준이 없으나, 한글T_EX글꼴이름 페이지에서 이 문제가 논의되었다. 여기서는 ‘비표준’이라는 점을 고려하여 폰트 명칭을 my로 시작하기로 하되, hangul-ucs의 글꼴 이름 작성 관행에 따라 o를 앞에 붙이기로 한다.¹¹⁾

한글L^AT_EX에서 이 폰트들을 사용하기 위해 명조에 mj, 돋움에 gt, 장식글자에 gr을 할당하기로 하고, 굴림체를 위해서 gl을 쓰기로 한다.

그 다음, 보통 두께를 나타내는 m, 두꺼운 글씨를 나타내는 b, 기울인 글씨(oblique)를 나타내는 o를 각각 필요한 폰트 이름에 붙이기로 하면, 그 결과는 다음 표 4와 같다. 기울인 글꼴은 보통 두께에만 붙이기로 하였으므로 mo만을 만든다. 양재난초M을 쓰기로 한 장식 글자는 두꺼운 글꼴과 기울인 글꼴을 정하지 않았다.

3.2 TFM 텍 폰트 파일 얻기

L^AT_EX에서 폰트를 사용하려면 TFM이라는 T_EX Font metric을 생성해주어야 한다. 트루타입의 tfm을 추출해주는 유틸리티가 ttf2tfm이다.

10) H^AL^AT_EX에서는 KS-HL^ATeX.sfd와 UKS-HL^ATeX.sfd 두 종류의 sfd 파일이 필요하다.

11) hangul-ucs의 발전과정에서 정착된 관행으로서, o가 앞에 붙는 것은 그것이 유니코드임을 의미한다.

현재 우리가 사용하려 하는 ttf2tfm은 버전 1.5 이상이어야 한다. 현재 대부분의 배포판에 1.5 버전이 포함되어 있으며, 이것은 2002년 KTUG에서 이루어진 중요한 기여 중의 하나이다.

한글 \LaTeX 은 한글 폰트를 여러 개의 하위폰트들로 쪼개어서 활용하는 방식, 이른바 “subfont 방식” 이므로, 하나의 ttf 파일에서 많은 TFM 파일들을 추출해야 한다.

각 폰트 파일에 대하여 다음과 같은 명령을 차례로 실행한다. 이 작업은 배치파일을 만들거나 셸 스크립트를 이용하면 훨씬 쉽게 작업할 수 있을 것이다.¹²⁾

```
ttf2tfm MSBatang.ttf -q -P 3 -E 1 omymjm@UCS2@ >> ttfonts.map
ttf2tfm MSBatang.ttf -q -s 0.167 -P 3 -E 1 omymjmo@UCS2@ >> ttfonts.map
ttf2tfm h2mjrb.ttf -q -P 3 -E 1 omymjb@UCS2@ >> ttfonts.map
ttf2tfm MSDotum.ttf -q -P 3 -E 1 omymgtm@UCS2@ >> ttfonts.map
ttf2tfm MSDotum.ttf -q -s 0.167 -P 3 -E 1 omymgtmo@UCS2@ >> ttfonts.map
ttf2tfm hmfbg.ttf -q -P 3 -E 1 omymgtb@UCS2@ >> ttfonts.map
ttf2tfm MSGulim.ttf -q -P 3 -E 1 omymglm@UCS2@ >> ttfonts.map
ttf2tfm MSGulim.ttf -q -s 0.167 -P 3 -E 1 omymglmo@UCS2@ >> ttfonts.map
ttf2tfm h2drrb.ttf -q -P 3 -E 1 omymglb@UCS2@ >> ttfonts.map
ttf2tfm ynch05.ttf -q -P 3 -E 5 omymgrm@UKS-UCS2@ >> ttfonts.map
ttf2tfm ynch05.ttf -q -s 0.167 -P 3 -E 5 omymgrmo@UKS-UCS2@ >> ttfonts.map
```

이 명령의 의미는 대강 다음과 같다.

먼저, 명조체의 보통 글꼴인 omymjm를 얻기 위해서 윈도바탕체 글꼴 MSBatang.ttf로부터 UCS2.sfd에 정의된 규칙을 따라서 tfm을 생성한다. 그러면 omymjmXX.ttf들이 생겨나는데, 예를 들면 한글 음절문자 영역의 tfm들은 omymjm_{ac}.tfm부터 차례로 생성된다.

트루타입 글꼴은 그 안의 문자 배열이 완성형 인코딩(EUC-KR)을 따르고 있는 것과 유니코드 인코딩을 따르고 있는 것 두 종류가 있다. 요즘은 대부분 유니코드 인코딩 폰트로 제작되지만 오래전에 만들어진 한글 폰트는 완성형이 많았다.¹³⁾

유니코드 인코딩 폰트에 대해서는 -P 3 -E 1을 옵션으로 지정하고 sfd를 UCS2.sfd로 지정한다. 반면 완성형 인코딩 폰트에 대해서는 -P 3 -E 5를 지정하고 sfd를 UKS-UCS2.sfd로 한다. 위의 예에서 양재난초는 완성형 폰트이고 다른 것은 유니코드 폰트이다.¹⁴⁾

그 결과를 ttfonts.map이라는 파일에 기록한다.¹⁵⁾

12) 13 페이지 4 절을 보라.

13) 완성형 인코딩 폰트에는 완성형 밖의 문자, 예컨대 “똥”, “떡” 같은 글자가 들어 있지 않으므로 이런 글자를 식자할 수 없다. 유니코드 인코딩이라 하더라도 이런 문자가 들어 있지 않은 폰트도 적지 않다.

14) ttf2tfm을 실행해보면 만약 폰트 인코딩이 맞지 않을 때 Pid와 Eid 값을 보여주면서 에러를 내므로, 처음에 잘못 지정했더라도 다시 시도할 수 있다.

15) dvipdfmx만을 이용할 목적일 경우 ttfonts.map을 굳이 생성할 필요는 없다. 그러나 이 파일을 만들어두면 ttf2pk에 의하여 PK 폰트 파일이 생성되게 할 수 있으므로 xdvi나 dvips를 사용할 수 있게 된다.


```
$ ttf2tfm MSBatang.ttf -q -P 3 -E 1 omymjm@UCS2@
```

그 다음, 명조체의 기울인 꼴(itshape)인 omymjmo를 얻기 위해서 역시 MSBatang.ttf로부터 TFM을 생성하게 한다. 이 때 slant 옵션을 명시한다(여기서는 .167).

```
$ ttf2tfm MSBatang.ttf -q -s 0.167 -P 3 -E 1 omymjmo@UCS2@
```

명조체의 굵은 글꼴의 upshape인 omymjb를 얻기 위해서, h2mjrb.ttf로부터 TFM을 얻는 같은 작업을 실행한다.

```
$ ttf2tfm h2mjrb.ttf -q -P 3 -E 1 omymjb@UCS2@
```

나머지도 이와 동일한 의미이다. 다만, 위의 일련의 명령에서 우리가 선택한 양재난초체는 유니코드 폰트가 아니라 완성형 폰트이므로, -E 옵션의 값을 5로 주고 sfd를 UKS-UCS2로 하여 글꼴 배열이 완성형임을 지정한다.¹⁶⁾

```
$ ttf2tfm ynch05.ttf -q -P 3 -E 5 omygrm@UKS-UCS2@
```

유니코드 폰트의 경우는 위에서 보인 대로 이 값이 1이다.

에러 없이 실행되면 수많은 TFM 파일들이 같은 디렉토리에 생겨나 있을 것이다.

3.3 FD 폰트 정의 파일 만들기

이제 L^AT_EX에서 이 폰트들을 이용하려면 폰트 정의(font definition) 파일을 작성해 주어야 한다. 폰트 정의 파일의 명칭은 아주 중요한데, 그 대략의 형식은 다음과 같다.

```
<font encoding><fontfamily name>.fd
```

한글L^AT_EX에서 사용하는 font encoding은 LUC 인코딩이다. 그리고, 여기서 fontfamily name이라고 하는 것은 실제 sub font들 자체인 omymjmac.tfm들 하나하나가 아니라, L^AT_EX이 이 일군의 폰트들을 일괄 참조하는 명칭을 말한다. 우리는 여기서 omymjXYY.tfm들을 mymj라는 이름으로 참조하도록 하자. 그러면 fd 파일의 이름은 LUCmymj.fd가 되어야 한다. 이 문제와 관련된 상세한 내용은 L^AT_EX 2_ε의 폰트 선택 스킴인 NFSS를 이해하여야 하는데, 필자가 쓴 “한글 글꼴 설치 사용법”¹⁷⁾이라는 문서를 (급한대로) 참조할 수 있다.

우리가 채택한 글꼴은 한자 부분을 포함하고 있는 것들이지만, 양재난초M은 한자 부분이 없다. 한글 ttf 폰트들 가운데는 한자 부분이 채워져 있지 않은 것이 많다는 점에 주의해야 한다. 이런 폰트는 ttf2tfm을 실행하였을 때 한자 부분에 해당하는 영역(plane)은 subfont tfm이 생성되지 않는다. 특히 두꺼운 글꼴로 보통 글꼴과 다른 폰트를 채택하는 경우, 한자 부분이 없는 글꼴을 선택하게 되면 한자를 \bfseries로 쓸 때 폰트 에러를 만날 것이다. 임의의 한글 글꼴을 사용하려 한다면 이 점에 주의해야 한다. 한자 부분이 없는 굵은 글꼴을 바탕글 서체로 채택하였을 때, 굵은

16) 따라서, 폰트 대응을 선정할 때 그 폰트의 성질을 미리 알아두면 편리하겠다.

17) http://faq.ktug.or.kr/faq/Karnes/2006-04?action=download&value=fntst_0430.pdf

```

\DeclareFontFamily{LUC}{mymj}{\hyphenchar\font\m@ne}

\DeclareFontShape{LUC}{mymj}{m}{n}{<->LUC * omymjm}{<->LUC * omymjmo}{<->LUC * omymjmo}{<->LUC * omymjm}{}
\DeclareFontShape{LUC}{mymj}{m}{sl}{<->LUC * omymjmo}{<->LUC * omymjmo}{<->LUC * omymjmo}{<->LUC * omymjm}{}
\DeclareFontShape{LUC}{mymj}{m}{it}{<->LUC * omymjmo}{<->LUC * omymjmo}{<->LUC * omymjmo}{<->LUC * omymjm}{}
\DeclareFontShape{LUC}{mymj}{b}{n}{<->LUC * omymjb}{<->LUC * omymjb}{<->LUC * omymjb}{<->LUC * omymjm}{}
\DeclareFontShape{LUC}{mymj}{bx}{n}{<->LUC * omymjb}{<->LUC * omymjb}{<->LUC * omymjb}{<->LUC * omymjm}{}

```

그림 1: LUCmymj.fd 파일의 내용

글꼴을 기본으로 사용하는 `\section` 명령의 인자 등에서 한자를 사용하면 안 된다. 한글 IAT_EX 은 한글용 글꼴과 한자용 글꼴을 별도로 지정할 수 있게 하고 있다.

폰트 정의 파일의 내용은, 예를 들어 LUCmymj.fd의 경우, 그림 1과 같다.

위의 내용을 참고하여, 다른 폰트의 FD 파일들도 만든다. 대부분 mj를 gt나 gr 등으로 치환하는 것으로 충분하다. 이렇게 하면 LUCmymj.fd, LUCmygt.fd, LUCmygl.fd와 LUCmygr.fd 등 네 개의 각 글꼴 가족별 폰트 정의 파일을 만들어 낼 수 있다.

3.4 map 파일의 작성

dvips driver는 트루타입을 처리하지 못한다. 그러므로 dvips용 map 파일 하나만 작성하여 updmap을 이용하여 일괄 설정하는 편리한 절차를 따를 수 없고, DVIPDFMx와 pdf T_EX 및 ttf2pk를 위한 map 파일들을 작성해주어야 한다. 그런 다음, dvipdfmx.cfg와 ttf2pk.cfg 파일의 내용을 수정해주어야 한다.

map 파일의 이름을 각각, cid-myttf.map, myttf-pdfTeX.map, myttf2pk.map이라고 하기로 하자.

cid-myttf.map 그림 2와 같은 내용으로 된 cid-myttf.map을 작성한다.¹⁸⁾

이 파일의 설정과 관련된 자세한 사항은 dvipdfmx의 관련 문서를 참고하라. 완성형 글꼴과 유니코드 글꼴의 설정이 다르다는 점에도 유의해서 보아야 한다.

DVIPDFMx는 ttc 파일도 다룰 수 있는데, 이 경우 map에 들어가는 행은 다음과 같은 형식을 갖는다.

```
omygsm@UCS2@ unicode :2:batang.ttc
```

18) 굵은 글꼴을 fakebold로 쓰기로 하였다면, 먼저 ttf2tfm으로부터 omymjb 폰트들을 같은 폰트로부터 얻은 다음 map에 다음과 같은 줄을 써준다.

```
omyymb@UCS2@ unicode :0:!MSBatang.ttf -b .1
```

```

omymjm@UCS2@    unicode    MSBatang.ttf
omymjmo@UCS2@  unicode    MSBatang.ttf -s .167
omymjb@UCS2@   unicode    h2mjrb.ttf
omygtm@UCS2@   unicode    MSGulim.ttf
omygtmo@UCS2@  unicode    MSGulim.ttf -s .167
omygtb@UCS2@   unicode    hmfbg.ttf
omyglm@UCS2@   unicode    MSGulim.ttf
omyglmo@UCS2@  unicode    MSGulim.ttf -s .167
omyglb@UCS2@   unicode    h2drrb.ttf
omygrm@UKS-UCS2@ KSCms-UHC-H ynch05.ttf
omygrmo@UKS-UCS2@ KSCms-UHC-H ynch05.ttf -s .167

```

그림 2: cid-myttf.map 파일의 내용

```

omymjm@UCS2@    <MSBatang.ttf PidEid=3,1
omymjb@UCS2@    <h2mjrb.ttf  PidEid=3,1
omygtm@UCS2@    <MSGulim.ttf  PidEid=3,1
omygtb@UCS2@    <hmfbg.ttf   PidEid=3,1
omyglm@UCS2@    <MSGulim.ttf  PidEid=3,1
omyglb@UCS2@    <h2drrb.ttf  PidEid=3,1
omygrm@UKS-UCS2@ <ynch05.ttf  PidEid=3,5

```

그림 3: myttf-pdftex.map 파일의 내용

여기서 :2:가 해당 폰트의 index 번호이다. 그러나 우리는 ttc를 모두 분리해서 쓰고 있으므로 font index를 따로 지정할 필요 없다.

완성형 글꼴에 지정하는 KSCms-UHC-H는 cmap의 명칭이다.¹⁹⁾

myttf-pdftex.map pdfTeX 1.40 이후 버전에서 map 파일은 subfont 문법을 쓸 수 있다. 우리의 경우는 그림 3과 같은 정도가 될 것이다.

안타깝지만 pdfTeX은 slanted 폰트를 직접 지원하지 않으며, fake-bold도 쓸 수 없다. 그러므로 slanted 폰트 관련 정의는 제외되었다.

19) 어떤 TeX 배포판에서는 필요한 cmap을 갖추고 있지 않을 수 있는데, 이럴 때는 Ghostscript의 Resource 폴더 아래에서 해당 cmap을 찾을 수 있을 것이다. 이것을 texmf 트리의 적당한 위치로 가져다두든가, 아니면 Ghostscript Resource 폴더를 cmap 탐색 위치로 texmf.cnf 설정을 바꾸든가 하는 조치를 취해야 할 수 있다.

표 5: DVIPDFMx와 pdfTeX의 트루타입 처리

	기본	획을 두겹게	기울임	가로폭	ttc
DVIPDFMx	가능	가능	가능	가능	가능
pdfTeX	가능	불가능	불가능	불가능	불가능

myttf2pk.map ttf2pk용 map은 위의 실행 과정에서 나온 ttfonts.map이다. 이 파일의 이름을 myttf2pk.map으로 변경한다.

설정 파일의 수정 이제 dvipdfmx.cfg 파일을 수정한다. 수정해야 할 dvipdfmx.cfg의 위치는 다음 명령으로 찾는다.

```
$ kpsewhich --format="other text files" --programe="dvipdfm" dvipdfmx.cfg
```

이 명령으로 찾아진 dvipdfmx.cfg 파일의 제일 마지막에 다음 한 줄을 추가하여 저장한다.

```
f cid-myttf.map
```

마찬가지 방법으로 ttf2pk.cfg를 찾아서, 다음 한 행을 추가한다.

```
map +myttf2pk.map
```

texmf 트리에 설치 이 map 파일들을 texmf 트리에 설치해두면 반복적으로 사용할 수 있다. 사용자 파일을 모아두는 texmf 트리²⁰⁾아래 다음과 같은 위치에 두는 것이 좋다.

```
[texmf]/fonts/map/dvipdfmx/myttf/cid-myttf.map
[texmf]/fonts/map/pdftex/myttf/myttf-pdftex.map
[texmf]/fonts/map/ttf2pk/myttf/myttf2pk.map
```

그런 다음 texhash(mktexlsr)를 실행한다.

3.5 테스트

위와 같이 준비가 모두 끝났으면, 이제 테스트 파일을 만들어서 시험해보자. 가장 간단한 테스트 파일을 그림 6과 같이 작성해보자.

이 파일을 test.tex으로 저장하고, 다음과 같이 실행한다.

20) KC2006의 경우는 %HOME%\texmf, Mac OS X의 texlive는 \$HOME/Library/texmf, Linux texlive에서는 \$HOME/texmf가 쓰인다.

```
#> latex test
#> dvipdfmx test
```

PDF 파일이 성공적으로 만들어지면 Adobe Reader와 같은 pdf viewer로 이 파일을 열어본다. 표 6 또는 이와 유사한 결과가 만들어지면 성공이다. 참고로, pdf \LaTeX 을 실행하면 이 가운데 기울인 글꼴은 pk 비트맵으로 처리된다.

위에서 우리가 만든 파일은 *.tfm, *.fd, 그리고 *.map이 전부였다.

4 ttf2kotexfont script

김도현 교수께서 2003년에 제작하였던 ttf2hlatexfont 스크립트를 변화된 상황에 맞게 수정한 새 판이 ttf2kotexfont이다. 이 스크립트를 이용하면 이 글에서 설명한 복잡한 작업을 간단히 명령행 한 줄로 일괄 처리할 수 있다.

ttf2kotexfont를 설치한 후, 다음과 같은 순서대로 하면 된다.

먼저 config 파일을 준비한다. 이 파일을 준비하기 위해서는 폰트 대응을 먼저 설계하여야 한다. 이 글에서 한 대로 하자면 대략 다음과 같은 내용이 될 것이다. 폰트 파일을 같은 폴더 또는 kpathsearch가 찾을 수 있는 위치(시스템 폰트 폴더를 읽을 수 있다면 거기에 있어도 된다)에 둔다.

```
FOUNDRY: my
FONTmj: m=MSBatang.ttf b=h2mjrb.ttf
FONTgt: m=MSDotum.ttf b=hmfbg.ttf
FONTgl: m=MSGulim.ttf b=h2drrb.ttf
FONTgr: m=yinch05.ttf
```

스크립트를 실행한다.

```
$ ttf2kotexfont -c config
```

실행이 끝나면 해당 폴더에 많은 .tfm 등이 생겨나 있을 것이다. 여기서 글자가 제대로 찍히는지 다음과 같이 테스트 파일을 컴파일해본다.

```
$ pdflatex testutf
```

또는

```
$ latex testutf
$ dvipdfmx testutf
```

만약 폰트 대응 등이 만족스럽고 이 폰트 세트를 지속적으로 사용하기를 원한다면 다음과 같이 자신의 local texmf tree에 등록하고 쓸 수 있다. 사용자의 local texmf

```

\documentclass[oldfontcommands,itemph]{oblivoir}
\SelectHfonts{mymj,mygt,*}{*}
\newcommand\gt{\SetAdhocFonts{mygt}{mygt}}
\newcommand\gl{\SetAdhocFonts{mygl}{mygl}}
\newcommand\gr{\SetAdhocFonts{mygr}{ungt}}

\begin{document}
\title{한글 글꼴을 시험합니다}
\author{}
\date{}
\maketitle

\begin{table}
\centering%
\caption{샘플 예제 파일}\label{exmpl}
\begin{tabular}{|c|c|c|c|}
\hline
글꼴 & 보통 & 굵음 & 기울임 \\ \hline
바탕 & 한글 & {\bf 한글} & {\it 한글} \\ \hline
\gt 돋움 & \gt 한글 & {\bf\gt 한글} & {\it\gt 한글} \\ \hline
\gl 굴림 & \gl 한글 & {\bf\gl 한글} & {\it\gl 한글} \\ \hline
\gr 장식 & \gr 한글 & & \\ \hline
\end{tabular}
\caption{한글 폰트 예제}\label{example}
\end{table}
\end{document}

```

표 6: 한글폰트예제

글꼴	보통	굵음	기울임
바탕	한글	한글	<i>한글</i>
돋움	한글	한글	<i>한글</i>
굴림	한글	한글	<i>한글</i>
장식	한글		

tree는 T_EX 배포판에 따라 조금씩 다른데, 이 글에서는 KC2006을 기준으로 하기로 하자. 이 때는 %HOME%\texmf가 된다.

```
$ ttf2kotexfont -c config -i %HOME%\texmf
```

여기서 약간의 경고가 나오지만, 무시하고 그냥 실행한다.

KC2006 시스템은 %HOME%\texmf 트리가 맨 먼저 읽는 texmf root가 아니다. 그러므로 .cfg 파일들을 최우선 texmf tree 아래로 이동시켜주는 작업을 해야 한다.²¹⁾ 이동은 다음과 같이 하면 된다.

```
%HOME%\texmf\dvipdfm\config\dvipdfmx.cfg
-> [TeX]\share\texmf-local\dvipdfm\config\dvipdfmx.cfg
%HOME%\texmf\ttf2pk\base\ttf2pk.cfg
-> [TeX]\share\texmf-local\ttf2pk\base\ttf2pk.cfg
```

이제 texhash 후에 해당 폰트를 사용할 수 있다.

```
$ mktexlsr
```

5 마치는 말

위의 절차를 따르면 어떤 임의의 ttf라도 한글L^AT_EX에서 쓰도록 할 수 있다. 그러나 아무리 ttf를 잘 사용할 수 있다 해도, 문서의 품위는 궁극적으로 글쓴이에게 달려 있다.

양재난초를 장식 글씨체로 쓰면 어떻게 되는지 아래에 예를 하나 드는 것으로 이 글을 마친다.²²⁾

내 사랑하리 시월의 강물을
 석양이 질어가는 푸른 모래톱
 지난날 가졌던 슬픈 여정들을, 아득한 기대를
 이제는 홀로 남아 따뜻이 기다리리.

지난 이야기는 해서 무엇하리
 두견이 우는 숲 새를 낚아서
 낮은 돌담에 흐르는 달빛 속에
 울리던 木桴소리 목금소리 목금소리.

21) 이것은 KC2006 설계상의 특징이다. 이 이동 작업이 번거롭기는 하지만 kcupdate가 cfg 파일들을 관리하고 있으므로 사용자가 직접 변경을 가하는 것은 약간 번거로워도 오히려 자신이 무슨 일을 하고 있는지 자각할 수 있으므로 도움이 되지 않는 것은 아니라고 생각한다.

22) 황동규의 시, “시월”의 일부. 장식 글씨체의 한자 부분은 돋움체의 것을 가져다 썼다.