

# 한글 $\text{\LaTeX}$ 에서 임의의 Truetype 한글 폰트 사용하기

김강수      주철

2003년 2월 8일

## 목 차

<b>1. 들어가는 말</b>	<b>3</b>
1.1 제한 .....	3
1.2 많은 글꼴? .....	4
<b>2. 준비</b>	<b>5</b>
2.1 글꼴의 대응 .....	5
2.2 작업 준비 .....	6
<b>3. 절차</b>	<b>7</b>
3.1 $\text{\TeX}$ 폰트 이름 .....	7
3.2 TFM 텍 폰트 파일 얻기 .....	8
3.3 FD 폰트 정의 파일 만들기 .....	9
3.4 STY 스타일 파일 만들기 .....	11
3.5 dvipdfmx를 위한 cid map 파일의 작성 .....	11
3.6 테스트 .....	12
<b>4. 마치는 말</b>	<b>14</b>
<b>부록</b>	<b>15</b>
<b>A 한글 TTF 폰트 사용에 관한 토론들</b>	<b>15</b>
<b>B texmf 트리에 설치하는 경우 파일의 위치</b>	<b>17</b>

	2
<b>C</b> 리눅스에서 <b>freetype</b> 및 <b>ttf2tfm</b> 설치	<b>18</b>
<b>D</b> <b>ttf.sh</b> 를 이용하는 방법	<b>19</b>
<b>E</b> <b>FD</b> 파일의 내용	<b>20</b>

## 감사의 말

이 글은 KTUG 문서화 작업팀의 공동 작업의 결과물이다. 초고는 김강수가 작성하였고, 주철은 리눅스에서의 설치 및 스크립트 작성을 담당하였다.

한글 트루타입 폰트를  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 에서 사용할 수 있게 된 것은 거의 전적으로 조진환, 박원규, 김도현 님 등 몇 분의 헌신적인 개발자들의 업적이다. 그 과정은 부록 A에 간략히 요약해 두었다. 이 분들이 개발하고 개선한 방법을 누구나 적용해볼 수 있도록 작성한 안내서가 이 문서이다. 개발자들에게 감사드린다.

이 길지 않은 글을 작성하는 데도 많은 분의 도움이 필요했다. 이주호 님은 초고를 읽고 문맥과 표현을 교정해주었고 내용에 대해서 유익한 의견을 제시해주셨다. 이기황 님은  $\text{texmf}$  트리의 위치에 관련된 내용을 추가할 것을 조언해주셨다. 두 분께 감사드린다.

## 1. 들어가는 말

KTUG<sup>1</sup>가 활동을 시작한 이후, 한글  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  사용환경은 몰라보게 달라졌다. 주요한 점들만 짚아보아도,

- 플랫폼에 관계없이 쉽고 간편하게  $\text{T}_{\text{E}}\text{X}$  시스템을 설치하고 운영할 수 있게 되었다. KTUG-MiKTeX이나 cygwin-teTeX과 같은 윈도우상의 설치 지원은 물론이고, cvs 서버를 이용하여 한글 관련 패키지들을 언제라도 다운받고 갱신할 수 있게 된 것이다.
- 한글 트루타입 폰트를 HLaTeX 패키지 또는 CJK 패키지와 함께 사용할 수 있게 됨으로써 폰트의 제약을 크게 극복하게 되었다.
- 모든 한글, 즉 EUC-KR 범위 밖의 한글은 물론이고, 한글 고문자도 처리할 수 있는 길이 열렸다.

이 글은 그 가운데 트루타입 폰트를 한글 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  패키지와 함께 사용하는 방법에 관한 간단한 사용자 안내서(User's Guide)이다.

### 1.1 제한

현재  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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.

<sup>1</sup>한글  $\text{T}_{\text{E}}\text{X}$  사용자 그룹. Korean TeX Users Group. <http://www.ktug.or.kr>.

1. 한글 패키지는 한글 $\text{\LaTeX}$ (HLaTeX) 패키지를 이용한다.
2. 입력 인코딩은 완성형 한글(KS X 1001, EUC-KR)로 한정한다. 따라서, 이 문서가 제시하는 방법에 의하여 표현할 수 있는 글자는 ‘완성형 한글’의 범위로 제한된다.<sup>2</sup>
3. 최종 출력은 PDF로 한다. DVI 출력물을 얻을 수는 있지만, 그것을 `dvipdfmx`로 처리하는 경우만을 설명한다. 따라서, 여러 종류의 DVI 처리기나, 그들이 요구하는 PK 폰트파일의 생성에 대해서는 다루지 않는다.<sup>3</sup>
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.

## 1.2 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?

$\text{\LaTeX}$ 은 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$ 에서 사용되는 글꼴은 `\rm`, `\sf`, `\tt` 이 세 종류뿐이다. 강조를 위해서 기울임(이탤릭) 처리가 추가되거나 **boldface**, **SMALL CAPITAL**을 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.

<sup>2</sup>UTF-8 입력 인코딩을 이용하여 “모든 한글”을 처리하는 방법에 관한 사용자 안내 문서는 KTUG 문서화 팀에 의하여 현재 준비 중이다.

<sup>3</sup>그러나 사실은, 이 문서에서 제시한 방법을 따라 TTF 글꼴을 사용가능하게 만들고 `ttf2pk` 1.5 이후 버전이 정상적으로 설치되어 있다면, `xdvi`나 `dvips`를 실행할 때 PK 폰트 파일이 자동적으로 생성된다. 이 글은 그 문제에 대해서 “다루지 않을” 뿐이고, 사용자는 보다 손쉽게 다른 DVI 처리기들을 활용할 수 있을 것이다. 특히, KTUG-MiKTeX을 설치 운영하고 있다면 거기에 포함된 DVI Previewer인 `yap`은 TTF 글꼴을 직접 화면에 보여줄 수 있다.

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 좋다.

한글 윈도가 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.

**T<sub>E</sub>X** 시스템 윈도 플랫폼의 경우, KTUG-MiK<sub>T</sub>E<sub>X</sub>을 권장한다. 다른 임플리멘테이션이라도 문제는 없다. KTUG-MiK<sub>T</sub>E<sub>X</sub>을 권장하는 이유는 이것을 설치하였을 때 `ttf2tfm`이나 `dvipdfmx` 최신판이 자동으로 설치되기 때문이다.

한글<sub>A</sub>T<sub>E</sub>X 한글<sub>E</sub>T<sub>E</sub>X 0.991 이상이 설치되어 있어야 한다.

`ttf2tfm`, `dvipdfmx` 이 두 프로그램은 KTUG-MiK<sub>T</sub>E<sub>X</sub>을 설치하면 바로 설치되어 있으므로 별도의 작업이 필요없다. 이 문서는 `ttf2tfm` 1.5 이상을 요구하며, `dvipdfmx`는 최신 버전을 권장한다. 만약 다른 **T<sub>E</sub>X** 시스템을 운용하고 있다면, 윈도의 경우 KTUG-MiK<sub>T</sub>E<sub>X</sub>을 설치하거나, `ttf2tfm`과 `dvipdfmx`를 컴파일하여야 한다.<sup>4</sup>

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

### 2.1 글꼴의 대응

한글 폰트의 사용에 있어서 문제가 되는 것은, 조진환 님이 지적하신 바와 같이,<sup>5</sup> 볼드체(두꺼운 글꼴)가 없다는 점이다. `dvipdfmx`에서 볼드체를 처리해줄 수 있

<sup>4</sup>예를 들어, cygwin-teTeX에서 `dvipdfmx`를 설치하는 문제는 다음 페이지에서 도움을 얻을 수 있다. <http://project.ktug.or.kr/cygwin-teTeX/>. 그리고, cygwin-teTeX에서 `freetype`과 `ttf2tfm`을 설치하는 방법은 이 문서의 부록 C에 제시된 리눅스에서의 설치방법을 참고하라.

<sup>5</sup><http://www.ktug.or.kr/jsboard/read.php?table=setup&no=1005&page=1>

글꼴	보통 서체	두꺼운 서체
바탕글꼴	윈도새바탕	한양태명조
돋움글꼴	윈도돋움	휴먼태고딕
굴림글꼴	윈도굴림	한양태등근고딕
장식글꼴	양재난초M	

표 1: 글꼴 대응

지만 별도의 글꼴을 사용하지 않는 한 글꼴을 포함시킬 수가 없어서 “불완전한 PDF”가 얻어진다. 이 PDF는 Adobe Acrobat Reader로 볼 때는 문제없으나, gv 등으로 볼 때는 에러를 내는 경우가 생긴다.<sup>6</sup>

그러므로, 볼드체 글꼴은 따로 설정해주는 것이 좋다. 예를 들면, 명조체에는 견명조체를 그 볼드체로 하는 등의 방법이다. 이 글에서는 비교적 쉽게 구할 수 있는 글꼴을 사용하여, 다음 표 1과 같이 대응시켜 사용해보기로 한다.<sup>7</sup>

윈도 돋움과 윈도 굴림은 하나의 TTC 파일<sup>8</sup>에 포함되어 있으므로, 이 둘을 분리해야 할 필요가 있다.<sup>9</sup> 이를 위해서 `breakttc.exe`라는 유틸리티가 필요하다.<sup>10</sup> `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 `gulim.ttf`로, `FONT02.ttf`를 `dotum.ttf`로 이름을 바꾸어서 사용한다.

## 2.2 작업 준비

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

<sup>6</sup>주철 님의 리포트

<sup>7</sup>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 20를 보라.

<sup>8</sup>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.

<sup>9</sup>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. 이 글에서는 글꼴을 분리하여 사용하는 쪽을 선택하였다.

<sup>10</sup>도스용 프로그램인데, MS에서 제공하는 TTSDK(`ttSDK.zip`)에 포함된 것. 리눅스에서는 `ttc2ttf`라는 유틸리티가 있다. 다음 글을 참고하라. <http://mail.mizi.com/pipermail/devel/2002-January/004279.html>

폰트 이름	폰트 파일 이름
윈도새바탕	nbatang.ttf
한양태명조	h2mjrb.ttf
윈도돋움	dotum.ttf
휴먼태고딕	hmfbg.ttf
윈도굴림	gulim.ttf
한양태등근고딕	h2drrb.ttf
양재난초M	ynch05.ttf

표 2: 폰트 파일 목록

1. 필요한 TTF 파일들을 이 디렉토리로 복사한다. 위의 폰트 대응에 따를 때 필요한 파일의 목록은 표 2와 같다.
2. 같은 디렉토리에 `KS-HLaTeX.sfd`와 `UKS-HLaTeX.sfd` 파일을 복사한다. 이 두 파일은 `dvipdfmx`를 설치하면 `$TEXMF/dvipdfm/sfd` 디렉토리<sup>11</sup>에 있다.
3. 같은 디렉토리에 `dvipdfmx`의 `dvipdfmx.cfg` 파일을 복사한다. 임시로 작업하기 위해서이다. `$TEXMF/dvipdfm/config`에 있는 파일을 이곳으로 복사해둔다.<sup>12</sup>

### 3. 절차

#### 3.1 TeX 폰트 이름

각 폰트 명칭에 해당하는 TeX 폰트 이름을 정해주어야 한다. 이 폰트 명칭은 일정한 일관성이 있으면 좋은데, 아직까지 표준이 없다. 이전에 KTUG 메일링리스트에서 조진환 님이 제안하신 바가 있지만<sup>13</sup> 그 이후로 진척이 없는 상태이다. 그러므로 여기서는 ‘비표준’이라는 점을 고려하여 폰트 명칭을 `my`로 시작하기로 한다.

한글 $\text{\LaTeX}$ 에서 이 폰트들을 사용하기 위해 명조에 `mj`, 돋움에 `gt`, 장식글자에 `gr`를 할당하기로 하고, 굴림체를 위해서 `gl`을 쓰기로 한다.

<sup>11</sup>KTUG-MiKTeX을 `C:\texmf`에 설치한 경우 `C:\texmf\vipdfm\sfd`

<sup>12</sup>이전의 `dvipdfm-cjk`에서는 `dvipdfm`의 경우와 같은 `config` 파일을 이용했으나, `dvipdfmx`의 설정파일은 `dvipdfmx.cfg`이므로 주의하라. 실행파일의 이름도 `dvipdfmx`이다.

<sup>13</sup><http://list.kldp.org/pipermail/ktug/2001Nov/0068.html>

폰트	보통	두꺼운	기울인
명조	mymj	mymjb	mymjo
고딕	mygt	mygtb	mymjo
굴림	mygl	myglb	myglo
장식	mygr		

표 3: T<sub>E</sub>X 폰트 이름

그 다음, 두꺼운 글씨를 나타내는 **b**, 기울인 글씨(oblique)를 나타내는 **o**를 각 필요한 폰트 이름에 붙이기로 하면, 그 결과는 다음 표 3과 같다. 양재난초M을 쓰기로 한 장식 글자는 두꺼운 글꼴과 기울인 글꼴을 정하지 않았다.

### 3.2 TFM 텍 폰트 파일 얻기

L<sup>A</sup>T<sub>E</sub>X에서 폰트를 사용하려면 TFM이라는 T<sub>E</sub>X Font로 변환하여야 한다. 이 일을 해주는 유틸리티가 `ttf2tfm`이다.

현재 우리가 사용하려 하는 `ttf2tfm`은 KTUG에서 논의된 사항이 반영된 버전<sup>14</sup>으로서 버전 1.5 이상이어야 한다. 한글L<sup>A</sup>T<sub>E</sub>X은 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 배치 파일을 만들거나 셸 스크립트를 이용하면 훨씬 쉽게 작업할 수 있을 것이다.<sup>15</sup>

```
ttf2tfm nbatang.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mymj@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm nbatang.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 mymjo@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm h2mjrj.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mymjb@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm h2mjrj.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 mymjb@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm dotum.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mygt@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm dotum.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 mygto@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm hmfbg.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mygtb@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm hmfbg.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 mygtbo@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm gulim.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mygl@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm gulim.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 myglo@UKS-HLaTeX@ >> ttfonds.map
```

<sup>14</sup>즉, 합자(ligature)를 처리할 수 있는 -l과 -L 옵션이 작동하는 버전. 이에 대해서는 다음 글을 참고할 것. <http://www.ktug.or.kr/jsboard/read.php?table=tips&no=11>

<sup>15</sup>이 문서에 딸린 부수파일 묶음에 간편하게 이 작업을 할 수 있도록 하는 셸 스크립트 예제를 포함하였다. 부록 D를 참고하라. 리눅스나 cygwin 환경에서 실행할 수 있다.

```

ttf2tfm h2drrb.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 myglb@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm h2drrb.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 myglbo@UKS-HLaTeX@ >> ttfonds.map
ttf2tfm ynch05.ttf -q -l -P 3 -E 5 mygr@KS-HLaTeX@ >> ttfonds.map
ttf2tfm ynch05.ttf -q -s 0.167 -l -P 3 -E 5 mygro@KS-HLaTeX@ >> ttfonds.map

```

이 명령의 의미는 대강 다음과 같다.

먼저, 명조체의 보통 글꼴인 `mymj`를 얻기 위해서 새바탕체 글꼴 `nbatang.ttf`로부터 `KS-HLaTeX.sfd`에 정의된 규칙을 따라서 TFM을 생성한다. 이 때 글꼴이 유니코드 배열을 가지고 있는 경우 `UKS-HLaTeX.sfd` 배열에서 `KS-HLaTeX.sfd`의 순서로 subfont를 생성하라는 옵션 `-L`을 지정한다. 그 결과를 `ttfonds.map`이라는 파일에 기록한다.<sup>16</sup>

```
ttf2tfm nbatang.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mymj@UKS-HLaTeX@
```

그다음, 명조체의 기울인 꼴(`itshape`)인 `mymjo`를 얻기 위해서 역시 `nbatang.ttf`로부터 TFM을 생성하게 한다.

```
ttf2tfm nbatang.ttf -q -s 0.167 -L KS-HLaTeX.sfd -P 3 -E 1 mymjo@UKS-HLaTeX@
```

명조체의 굵은 글꼴의 `upshape`인 `mymjb`를 얻기 위해서, `h2mjrb.ttf`로부터 TFM을 얻는 같은 작업을 실행한다.

```
ttf2tfm h2mjrb.ttf -q -L KS-HLaTeX.sfd -P 3 -E 1 mymjb@UKS-HLaTeX@
```

나머지도 이와 동일한 의미이다.

다만, 위의 일련의 명령에서 우리가 선택한 양재난초체는 유니코드 폰트가 아니라 완성형 폰트이므로, `-E` 옵션의 값을 5로 주고 글꼴 배열이 완성형임을 지정한다. 이 경우 `-l` 옵션만을 부여한다.<sup>17</sup> 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 `LATEX`에서 이 폰트들을 이용하려면 폰트 정의(font definition) 파일을 작성 해주어야 한다. 한글`LATEX`의 경우 같은 폰트에 대해서 세 종류의 `.fd` 파일이 필요한데, 각각 한글(`w`), 심볼(`s`), 한자(`h`) 폰트정의들이다. 이 폰트 정의 파일들의 명칭은 표 4와 같다.

<sup>16</sup>`dvipdfmx`만을 이용할 목적일 경우 `ttfonds.map`을 굳이 생성할 필요는 없다. 그러나 이 파일을 만들어두면 `ttf2pk`에 의하여 PK 폰트 파일이 자동으로 생성되게 할 수 있으므로 `xdvi`나 `dvips`를 사용할 수 있게 된다.

<sup>17</sup>따라서, 폰트 대응을 선정할 때 그 폰트의 성질을 미리 알아둘 필요가 있을 것이다.

글꼴	한글	심볼	한자
명조(mymj)	Hwmymj.fd	Hsmymj.fd	Hhmymj.fd
돋움(mygt)	Hwmygt.fd		
굴림(mygl)	Hwmygl.fd	Hsmzgl.fd	Hhmzgl.fd
장식(mygr)	Hwmygr.fd		

표 4: FD 파일의 명칭

```

\DeclareFontFamily{H}{wmyj}{\hyphenchar\font@m@ne}

\DeclareFontShape{H}{wmyj}{m}{n}{<->mymj}{}
\DeclareFontShape{H}{wmyj}{m}{sl}{<->mymjo}{}
\DeclareHFontShape{H}{wmyj}{m}{it}{<->hssub*wmyj/m/sl}{}

\DeclareFontShape{H}{wmyj}{b}{n}{<->mymb}{}
\DeclareFontShape{H}{wmyj}{b}{sl}{<->mymbo}{}
\DeclareHFontShape{H}{wmyj}{b}{it}{<->hssub*wmyj/b/sl}{}

```

그림 1: Hwmymj.fd 파일의 내용

이 표에서 돋움과 장식의 심볼, 한자 글꼴정의는 비워두었는데, 여기서는 이 두 곳에 각각 굴림체와 명조체의 심볼 및 한자를 가져다쓰기로 한다.

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 `ttf2tfm`을 실행하였을 때 한자 부분에 해당하는 11-29번 subfont가 생성되지 않는다. 특히 두꺼운 글꼴로 보통 글꼴과 다른 폰트를 채택하는 경우, 한자 부분이 없는 글꼴을 선택하게 되면 한자를 `\bfseries`로 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 글꼴을 기본으로 사용하는 `\section` 명령의 인자 등에서 한자를 사용하면 안 된다.

폰트 정의 파일의 내용은, 예를 들어 `Hwmymj.fd`의 경우,<sup>18</sup> 그림 1과 같다.

위의 내용을 참고하여, 다른 폰트의 FD 파일들도 만든다. 대부분 `mj`를 `gt`나 `gr` 등으로 바꾸는 것으로 충분하다.

이 폰트 정의 파일을 보다 손쉽게 만드는 스크립트를 박원규 님이 작성한 것이

<sup>18</sup>`mymj` 폰트의 각 폰트 정의 파일들의 내용은 부록 E를 참고하라.

```

%% myttffont.sty
%%
\ifx\clearocplists\undefined
\MapHangulFamily{mj}{mymj,mymj,mymj}
\MapHangulFamily{gt}{mygt,mygl,mygl}
\MapHangulFamily{gl}{mygl,mygl,mygl}
\MapHangulFamily{gr}{mygr,mymj,mymj}

\DeclareErrorHFont{H}{\mjdefault}{m}{n}{10}
\DeclareHFontSubstitution{H}{\mjdefault}{m}{n}
\fi

```

그림 2: myttffont.sty 파일의 내용

있는데, 주철 님이 개선한 것을 이 글의 부수파일로 제공한다. 이 글의 부록 **D**를 참고하라.

### 3.4 STY 스타일 파일 만들기

이제 이 폰트를  $\text{\LaTeX}$ 에서 사용하기 위한 스타일 파일을 만든다.<sup>19</sup> 스타일의 이름은 myttffont.sty로 하기로 하고, 그 내용을 그림 2와 같이 써 넣는다.

이 스타일이 하는 일은, mj, gt, gl, gr 글꼴을 설정하는 것인데, 위의 스타일에서는 gr의 심볼과 한자를 mj에서, gt의 심볼과 한자를 gl에서 가져오도록 했다.

### 3.5 dvipdfmx를 위한 cid map 파일의 작성

최종 출력물을 PDF로 얻기 위해서 우리는 dvipdfmx를 사용할 것이다. 그러므로, 필요한 map 파일을 작성하고 dvipdfmx.cfg 파일을 수정해주어야 한다.

map 파일의 이름은 myttffont.map으로 하기로 한다. 그림 3과 같은 내용으로 된 myttffont.map을 작성한다.<sup>20</sup>

이 파일의 설정과 관련된 자세한 사항은 dvipdfmx의 관련 문서를 참고하라.

<sup>19</sup>굳이 스타일을 만들지 않아도, 같은 내용을 preamble에 추가하는 것으로 동일한 효과를 얻을 수 있다.

<sup>20</sup>굵은 글꼴을 별도로 설정하지 않았다면, 예를 들어 명조체의 굵은 글꼴을 위해서 다음과 같이 설정하여야 한다. mymjb@KS-HLaTeX@ KSCms-UHC-H :0:!nbatang.ttf,Bold

mymj@KS-HLaTeX@	KSCms-UHC-H	:0:nbatang.ttf
mymjo@KS-HLaTeX@	KSCms-UHC-H	:0:nbatang.ttf -s .167
mymjb@KS-HLaTeX@	KSCms-UHC-H	:0:h2mjrb.ttf
mymjbo@KS-HLaTeX@	KSCms-UHC-H	:0:h2mjrb.ttf -s .167
mygt@KS-HLaTeX@	KSCms-UHC-H	:2:gulim.ttc
mygto@KS-HLaTeX@	KSCms-UHC-H	:2:gulim.ttc -s .167
mygtb@KS-HLaTeX@	KSCms-UHC-H	:0:hmfbg.ttf
mygtbo@KS-HLaTeX@	KSCms-UHC-H	:0:hmfbg.ttf -s .167
mygl@KS-HLaTeX@	KSCms-UHC-H	:0:gulim.ttc
myglo@KS-HLaTeX@	KSCms-UHC-H	:0:gulim.ttc -s .167
myglb@KS-HLaTeX@	KSCms-UHC-H	:0:h2drrb.ttf
myglbo@KS-HLaTeX@	KSCms-UHC-H	:0:h2drrb.ttf -s .167
mygr@KS-HLaTeX@	KSCms-UHC-H	:0:ynch05.ttf
mygro@KS-HLaTeX@	KSCms-UHC-H	:0:ynch05.ttf -s .167

그림 3: myttffont.map 파일의 내용

주의할 점은, 우리가 `gulim.ttc`에서 얻어낸 `gulim.ttf`와 `dotum.ttf`를 `texmf` 트리 아래에 두거나 Windows 폰트 등록을 하고 그대로 사용할 것이라면 5,6번 줄의 `:2:gulim.ttc` 부분을 `:0:dotum.ttf`로 바꾸어도 상관없다. 9,10번 줄의 `gulim.ttc`에 대해서도 마찬가지로 `:0:gulim.ttf`라고 쓰면 된다. 그러나, 일반적으로 윈도상에서 `dvipdfmx`를 돌리는 것이라면 위와 같이 정의하는 것이 더 좋을 것이다. 만들어지는 PDF에 폰트를 포함(embed)하지 않으려면 `:0:!nbatang.ttf`와 같이 폰트 이름 앞에 `!`를 추가하는 방식으로 정의한다.

이제 같은 디렉토리에 있는 `dvipdfmx.cfg` 파일을 수정한다.<sup>21</sup>

`dvipdfmx.cfg` 파일의 제일 마지막에 다음 한 줄을 추가한다.

```
f myttffont.map
```

### 3.6 테스트

위와 같이 준비가 모두 끝났으면, 이제 테스트 파일을 만들어서 시험해본다. 가장 간단한 테스트 파일을 그림 4와 같이 작성해보자.

이 파일을 `test.tex`으로 저장하고, 다음과 같이 실행한다.

<sup>21</sup>물론 시스템의 `dvipdfmx.cfg` 파일을 수정해도 되지만, 여기서는 특정 작업 디렉토리에 이 파일을 복사하고 수정하는 쪽을 권장한다.

```

\documentclass{article}
\usepackage{hangul}
\usepackage{myttffont}

\newcommand\gr{\hfontfamily{gr}}
\newcommand\gl{\hfontfamily{gl}}

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

\begin{table}
\centering%
\begin{tabular}{|c|c|c|c|}
\hline
글꼴 & 보통 & 굵음 & 기울임 \\ \hline
바탕 & 한글 & {\bf 한글} & {\it 한글} \\ \hline
\gt & 돌음 & \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}

```

그림 4: 샘플 예제 파일

```

#> latex test
#> dvi2pdf test

```

PDF 파일이 성공적으로 만들어지면 Acrobat Reader로 이 파일을 열어본다. 표 5가 만들어지면 성공이다.

글꼴	보통	굵음	기울임
바탕	한글	한글	한글
돋움	한글	한글	한글
굴림	한글	한글	한글
장식	한글		

표 5: 한글폰트예제

## 4. 마치는 말

위에서 우리가 만든 파일은 \*.tfm, \*.fd, \*.sty 그리고 \*.map이 전부였다. 이 파일들을 texmf 트리의 일부로 설치해두면 이 폰트 세트를 지속적으로 활용할 수 있다. 비록 표준은 아니지만, 만약 texmf 트리의 어떤 위치에 파일을 배치할 것인지에 대한 하나의 예를 부록 B에 들어 두었다.

이 글을 계기로, 한글 Truetype 글꼴 사용의 표준이 확립되기를 바라는 마음이다.

위의 절차를 따르면 어떤 임의의 TTF라도 한글<sup>22</sup>TeX에서 쓰도록 할 수 있다. 문제는 유용한 TTF 폰트가 충분히 확보될 수 있는가, 그리고 폰트를 적절하게 대응시켜서 쓰도록 디자인할 수 있는가 하는 것이지 TTF 폰트를 사용할 수 있는가 여부가 이제는 아니라고 생각한다. 아무리 TTF를 잘 사용할 수 있다 해도, 문서의 품위는 궁극적으로 글쓴이에게 달려 있다.

양재난초를 장식 글씨체를 쓰면 어떻게 되는지 아래에 예를 하나 드는 것으로 이 글을 마친다.<sup>22</sup>

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

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

<sup>22</sup> 황동규의 시, “시월”의 일부. 장식 글씨체의 한자 부분은 명조체의 것을 가져다 썼다.

# 부 록

## A. 한글 TTF 폰트 사용에 관한 토론들

한글 $\text{\LaTeX}$ 에서 한글 TTF 폰트를 사용 가능하도록 만드는 일은 KTUG가 만들어지던 초창기부터 논의가 진행되어, 많은 개선과 개발이 이루어졌다. KTUG의 이곳저곳에 이에 대한 글들이 남아 있다. 그 가운데 주요한 글들을 소개한다.

출발 한글 TTF 폰트의 사용에 관한 첫번째 시도는 조진환 님에 의하여 이루어졌다. TrueType 폰트에서 TFM을 추출한 다음 `ttf2pk`를 이용하여 PK 비트맵 폰트 파일을 얻어서 이용하는 방식이었다. KTUG로 통합되기 이전의 ChoF's TeX Archive<sup>23</sup>에 이 문제에 대한 상세한 글이 올라온 것은 1999년 11월의 일이다. 다음 글을 참고하라.

<http://free.kaist.ac.kr/cgi-bin/CrazyWWWBoard.cgi?db=Docs&mode=read&num=11&page=4&>

<http://free.kaist.ac.kr/cgi-bin/CrazyWWWBoard.cgi?db=Docs&mode=read&num=12&page=4&>

<http://free.kaist.ac.kr/cgi-bin/CrazyWWWBoard.cgi?db=Docs&mode=read&num=13&page=4&>

조진환 님은, 한글Windows의 글꼴을 `tfm` 형태로 추출하여 `msttf.zip`이라는 파일 묶음을 만들었다. 다음은 이것을 `fptex`의 `texmf` 트리에 복사하여 사용하도록 만든 파일묶음이다.

<http://doeun.pe.kr/site-cgi/ezboard/ezboard.cgi?db=latex&action=read&page=2&num=27&dbf=200008190000>

`ttf2pk` 및 `ttf2tfm`의 패치 조진환 님의 작업을 박원규 님이 조금 개선한 것이 KTUG 메일링리스트가 열릴 때 거의 첫 글로 올라오게 된다(2001/10). 이 글은 <http://list.kldp.org/pipermail/ktug/2001Oct/0001.html>에서 읽을 수 있다.

김도현 님이 PDF 파일을 만들 때 TrueType 폰트를 포함(`embed`)할 수 있다는 정보를 제공한 이후로, `ttf2pk`(`ttf2tfm`)은 `*.enc` 인코딩 파일을 함께 얻어 최종출력물을 PDF로 하는 방향으로 발전해갔다. 그 후, `ttf2tfm` 자체

<sup>23</sup><http://free.kaist.ac.kr/ChoF/>

에서 합자(ligature)를 처리하는 옵션을 추가하는 등 일련의 패치 작업이 조진환 님에 의해서 이루어졌고, 그 결과는 프로그램의 저자에 의해 받아들여져 현재의 `ttf2pk 1.5` 버전이 나오게 된 것이다. 트루타입의 사용 문제는 `ttf2tfm`과 `ttf2pk`의 패치를 통하여 사실상 거의 해결된 셈이다.

이 과정을 조진환 님이 요약해둔 글이 다음에 있다.

<http://www.ktug.or.kr/jsboard/read.php?table=tips&no=11>

**Truetype을 포함한 PDF : PdfTeX** 초창기 KTUG 메일링리스트에서 Truetype을 PDF에 포함(embed)하는 문제에 대한 논의가 이루어지고 난 다음, 그 논의를 정리하는 글이 김도현 님에 의하여 작성되었다. 백묵글꼴과 PdfTeX을 이용하는 방법에 관한 글이었는데, 적어도 이 문제에 관한 한 고전에 속한다고 생각된다.

<http://www.ktug.or.kr/jsboard/read.php?table=tips&no=7>

이 방법의 개요를 요약한 조진환 님의 글이 있다.

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=26>

이 방법을 이용하여 제작된 PDF 파일의 예는, 다음 문서이다.

<http://www.doeun.pe.kr/uselatex/ref-files/l4wp-kr/l4wp-kr.pdf>

**Truetype을 이용한 PDF : dvipdfm-kor** 트루타입의 사용 문제는 TTF 폰트를 활용하는 문제로서만 다루어지지 않고 고품위의 한글 PDF를 만드는 문제와 밀접하게 결부되어 발전하였다.<sup>24</sup> PDF 문서에 글꼴을 포함하지 않고 시스템에 있는 글꼴을 이용하도록 PDF를 작성하면 텍스트의 검색 추출이 가능하다는 것을 이용하여, 조진환 님은 `dvipdfm`의 패치를 시도하였다. 이 때부터 비로소 한글 검색이 가능한 PDF가 L<sup>A</sup>T<sub>E</sub>X으로 만들어진다. 2001년 12월의 일이었다. 다음 일련의 글들을 참고하라.

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=63>

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=69>

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=80>

이 패치는 `dvipdfm-kor` 프로젝트로 이어진다.

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=207>

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=218>

<http://www.ktug.or.kr/jsboard/read.php?table=ktugbd&no=262>

이 결과로 만들어진 `dvipdfm-kor`를 이용하여 만든 문서의 예는 다음과 같다.

<sup>24</sup>물론, 고품위 한글 PDF가 반드시 TTF 폰트를 사용하여야 한다는 것은 아니다.

<http://www.ktug.or.kr/doc/gfaq/gfaq.pdf>

조진환 님의 `dvipdfm-cjk` `dvipdfm-kor` 작업은 Shunsaku Hirata 님의 `dvipdfm-jpn`와 합쳐져서 `dvipdfm-cjk`<sup>25</sup> 프로젝트로 시작되었다. 이 프로젝트는 국제적 규모로 진행되는 것으로, 한/중/일에서 모두 관심이 매우 큰, 성공적인 프로젝트 가운데 하나이다. 현재는 한글—CJK문자 모두—이 포함된 PDF 파일을 만드는 “사실상의 표준(*de facto standard*)”으로 자리잡아가고 있다.

`dvipdfmx` `dvipdfm-cjk`는 또 한 단계 도약을 거쳐 `dvipdfmx`로 발전하였다.

대강 이러한 발전 단계를 거쳐, 이제는 트루타입을 비교적 자유롭게 활용할 수 있게 되었을 뿐 아니라, 훌륭한 PDF 출력을 얻을 수 있게 되었다.

## B. `texmf` 트리에 설치하는 경우 파일의 위치

우리가 만든 여러 부수파일들을 `texmf` 트리 아래 두려는 경우는 이 글에서는 권장하지 않는다. 그 대신 모든 파일을 작업 디렉토리(이 글의 경우 `C:\makefont`)에 두고 거기에서 작업하도록 하는 것을 염두에 두고 글을 작성하였다.

그러나, 경우에 따라서는 `texmf` 트리에 이 파일들을 설치한 후, 여러 문서를 같은 폰트 세트에 조판하고자 하는 경우가 있을 것이다. 그럴 때 어떻게 파일들을 배치하는가 하는 것은 TDS에 일치하기만 한다면 어떤 형태라도 상관없지만, 다음에 제시하는 예는 그 한 가지 방법이다.

다만, `texmf` 트리는 별도의 새로운 트리로 만들 것을 강력히 권장한다. 그렇게 하는 편이 나중에 이 파일들을 관리하기 쉬워질 것이라고 생각된다.<sup>26</sup>

```
$myttf-texmf/fonts/tfm/myttffont/*.tfm
```

```
$myttf-texmf/fonts/truetype/myttffont/*.ttf
```

<sup>25</sup><http://project.ktug.or.kr/dvipdfm-cjk/>

<sup>26</sup>별도의 `texmf tree`를 관리하는 방법은  $\TeX$  implementation에 따라 다르다. 예를 들어 `teTeX`이라면 새로운 트리를 `/home/user/myttf-texmf`라고 했을 때, 이 트리를 하나 만들고 `/usr/share/texmf/web2c/texmf.cnf`를 수정하여야 한다. 수정하는 방법은 다음과 같다. 먼저 `TEXMF` 변수를 하나 더 설정한다. 예를 들면 그것을 `MYTTFTEXMF`라고 하기로 하자. `texmf.cnf` 파일을 에디터로 열고, 여기서 다음 한 행을 추가한다. 대개 70행 부근에 이와 관련된 설정이 있을 것이다.

```
MYTTFTEXMF = $HOME/myttf-texmf
```

그런 다음 `TEXMF`의 현재 설정이 이 변수를 추가한다.

```
TEXMF = {$MYTTFTEXMF,!!$TEXMFMAIN}
```

`mktexlsr`를 실행하여 이 디렉토리에 `ls-R` 파일이 생성되는가를 본다.

```
$myttf-texmf/tex/latex/myttffont/myttffont.sty
$myttf-texmf/tex/latex/myttffont/*.fd
```

```
$myttf-texmf/dvipdfm/base/myttffont/myttffont.map
$myttf-texmf/dvipdfm/config/config
```

이 가운데, `config` 파일은 되도록 작업 디렉토리에 같이 두고 작업하는 쪽이 여러 모로 나을 것으로 생각한다. `xdvi` 등을 쓰려는 경우라면 `ttfonts.map`을 `myttf-ttfonts.map` 정도의 이름으로 `$myttf-texmf/ttf2tfm/base` 정도의 디렉토리에 옮겨놓고, 같은 디렉토리에 있는 `ttf2pk.cfg` 파일에 다음 한 줄을 추가한다.

```
map +myttf-ttfonts.map
```

그런 다음 `mktextlsr`과 같은 파일네임 데이터베이스 갱신 명령을 실행한다. `MiKTeX`에 서는 `MiKTeX Options`를 통해서 같은 일을 할 수 있을 것이다.

## C. 리눅스에서 freetype 및 ttf2tfm 설치

`ttf2pk(ttf2tfm)` 1.5 버전부터는 이미 조진환님의 패치가 적용되어 있다.<sup>27</sup>

`freetype`과 `ttf2pk(ttf2tfm)` 소스를 받아 컴파일하는 순서는 다음과 같다.

```
cvs -d:pserver:anonymous@cvs.freetype.org:/cvs/freetype login
[CVS password: anonymous]
cvs -z3 -d:pserver:anonymous@cvs.freetype.org:/cvs/freetype co freetype
cvs -z3 -d:pserver:anonymous@cvs.freetype.org:/cvs/freetype \
    co freetype1-contrib
cp -r freetype1-contrib freetype/
cd freetype
./configure --prefix=/usr
make
make install
cd freetype1-contrib/ttf2pk
./configure --prefix=/usr --with-kpathsea-lib=/usr/lib \
    --with-kpathsea-include=/usr/include/kpathsea
make
```

<sup>27</sup><http://www.ktug.or.kr/jsboard/read.php?table=tips&no=11>

`make install`

`prefix` 옵션이나 `kpathsea library path` 옵션은 적당히 바꾸어야 할 수도 있으므로, README 파일을 살펴보라.

## D. ttf.sh를 이용하는 방법

리눅스나 `cygwin` 시스템에서, 이 글에서 설명한 내용을 보다 간단하게 작업하도록 하는 스크립트를 주철 님이 작성하였다. 이 스크립트를 이용하는 경우의 작업 순서이다.<sup>28</sup>

1. 필요한 글꼴 대응을 정하고 글꼴을 특정 디렉토리에 가져다 두거나 심볼릭 링크를 건다. 여기서는 이 글에서 설명한 글꼴 쌍을 이용하는 방법을 예로 들어본다.
2. `ttf.sh`를 같은 디렉토리에 가져다둔다.
3. 다음 명령을 차례로 실행한다.
 

```
./ttf.sh nbatang.ttf mymj -B h2mjrb.ttf
./ttf.sh dotum.ttf mygt -B hmfbg.ttf
./ttf.sh gulim.ttf mygl -B h2drrb.ttf
./ttf.sh ynch05.ttf mygr
```
4. 이 디렉토리의 하위 디렉토리에 `mymj_fd`, `mymj_tfm` 등이 생겨나 있다. 같은 디렉토리에서 작업하기 위하여 이것들을 상위 디렉토리로 심볼릭 링크를 걸어준다.
 

```
ln -s ./my*_fd/* ./
ln -s ./my*_tfm/* ./
```
5. 같은 디렉토리에 파일 이름을 뒤에 붙이는 `map` 파일들이 있다. 파일들을 `config` 파일에 포함시킨다. 예를 들면,
 

```
f cid-nbatang.map
f cid-dotum.map
```

 등을 `config`에 추가하는 것이다. 또는 이 `map` 파일들을 모두 합쳐서 하나의 파일로 만들어도 된다.
6. 별도의 스타일을 만들지 않을 것이라면, 그림 2에서 설명한 내용을 `preamble`에 넣고 테스트 파일을 만들어서 시험해본다.

<sup>28</sup> 스크립트는 <http://doc.ktug.or.kr/ktug-doc/makettfavailable/ttf.sh>에서 받을 수 있다.

이 스크립트는 유니코드 폰트를 기본값으로 가정하고 예러가 발생하면 완성형 폰트 옵션으로 다시 시도한다.

## E. FD 파일의 내용

아래 예는 이 글의 `mymj` 폰트의 경우 각 `*.fd` 파일의 내용을 예시한 것이다. `mygt` 등 다른 폰트의 경우에도 마찬가지로 방법으로 `mj`를 `gt`로 바꾸어서 만들면 된다.

### Hwmymj.fd

```
\DeclareFontFamily{H}{wmymj}{\hyphenchar\font\m@ne}

\DeclareFontShape{H}{wmymj}{m}{n}{<->mymj}{}
\DeclareFontShape{H}{wmymj}{m}{sl}{<->mymjo}{}
\DeclareHFontShape{H}{wmymj}{m}{it}{<->hssub*wmymj/m/sl}{}

\DeclareFontShape{H}{wmymj}{b}{n}{<->mymjb}{}
\DeclareFontShape{H}{wmymj}{b}{sl}{<->mymjbo}{}
\DeclareHFontShape{H}{wmymj}{b}{it}{<->hssub*wmymj/b/sl}{}

```

### Hsmymj.fd

```
\DeclareFontFamily{H}{smymj}{\hyphenchar\font\m@ne}

\DeclareFontShape{H}{smymj}{m}{n}{<->mymj}{}
\DeclareFontShape{H}{smymj}{m}{sl}{<->mymjo}{}
\DeclareHFontShape{H}{smymj}{m}{it}{<->hssub*smymj/m/sl}{}

\DeclareFontShape{H}{smymj}{b}{n}{<->mymjb}{}
\DeclareFontShape{H}{smymj}{b}{sl}{<->mymjbo}{}
\DeclareHFontShape{H}{smymj}{b}{it}{<->hssub*smymj/b/sl}{}

```

### Hhmymj.fd

```
\DeclareFontFamily{H}{hmymj}{\hyphenchar\font\m@ne}

\DeclareFontShape{H}{hmymj}{m}{n}{<->mymj}{}

```

```
\DeclareFontShape{H}{hmymj}{m}{sl}{<->mymjo}{}  
\DeclareHFontShape{H}{hmymj}{m}{it}{<->hssub*hmymj/m/sl}{}  
  
\DeclareFontShape{H}{hmymj}{b}{n}{<->mymjb}{}  
\DeclareFontShape{H}{hmymj}{b}{sl}{<->mymjbo}{}  
\DeclareHFontShape{H}{hmymj}{b}{it}{<->hssub*hmymj/b/sl}{}  

```