글꼴 하우투

Donovan Rebbechi, *elflord@panix.com* <mailto:elflord@panix.com>

목 차

제 1 절 소개

1.1 이 문서가 있는 곳

이 문서는 *글쓴이의 웹페이지* `<http://pegasus.rutgers.edu/~elflord/font_howto>` 에 있다.

1.2 Rationale

글쓴이는 여기에서 글꼴에 대해 간단히 소개를 해 볼 작정이다. 이 글의 목적은 이제까지의 글꼴에 대한 글들이 건너뛰고 있던 빈 자리를 설명하는 것이다. 지금까지도 리눅스 환경에서의 글꼴에 대한 글은 여럿 있었지만, 그 가운데 알기 쉬운 것은 하나도 없었고, 모두 상대적으로 좁은 내용에 촛점을 맞추고 있었다는 것이 글쓴이의 느낌이다. 이 글의 목적은 글꼴 문제에 대해 획기적이고 새로운 이해를 제공하는 것이 아니라, 리눅스에서의 여러 가지 글꼴 문제를 해결하기 위한 출발점 구실을 할 수 있는 알기쉬운 자료를 제공하는 것이다.

1.3 감사의 글

Rod Smith에게 특별한 감사를 표한다. 그는 여러 가지 도움이 되는 제안들을 해주었고, pfm2afm을 고쳐주었다. Doug Holland에게도 감사한다. 그의 Font Deuglification HOWTO는 좋은 참고 자료가 되었다. 그리고 리눅스 공동체에게도 여러 가지 제안과 건설적인 비평에 대해 감사한다.

제 2 절 글꼴 101 – 글꼴에 대한 간단한 소개

2.1 글꼴의 종류

2.1.1 비트맵 글꼴

비트맵이란 점들로 이루어진 행렬이다. 비트맵 글꼴은 바로 점들로 이루어진 행렬로 표현된다. 때문에 비트맵 글꼴은 *장치 의존적 (device dependent)*이다. 즉 비트맵 글꼴은 특정한 해상도에서만 쓸모가 있다. 75 DPI의 스크린용 비트맵 글꼴은 1200 DPI의 해상도를 갖는 프린터에서도 여전히 75 DPI일 뿐이다.

비트맵 글꼴에는 두 가지가 있다. dvips에 의해 만들어지는 `pk` 글꼴와 같은 비트맵 프린터 글꼴과, X와 콘솔(console)에서 쓰이는 비트맵 스크린 글꼴이다. 비트맵 스크린 글꼴는 대개 `bdf`나 `pcf`라는 확장자를 갖는다. 비트맵 스크린 글꼴은 크기를 바꿀 수 없고 인쇄가 불가능하기 때문에 이런 점들이 문제가 되지 않는 터미널 윈도우나 콘솔, 텍스트 편집기 등에서 유용하다.

2.1.2 타입1 글꼴

타입1 글꼴 표준은 Adobe사에 의해 고안되었으며, Adobe의 포스트스크립트 표준이 타입1 글꼴을 지원한다. 유닉스에서는 전통적으로 인쇄에 관련된 일이라면 언제나 포스트스크립트 글꼴을 사용해 왔다. 이 때문에 리눅스에서도 타입1 글꼴은 잘 지원되고 있다. X와 고스트스크립트가 타입1 글꼴을 지원한다.

통상, 유닉스의 타입1 글꼴은 `afm` (adobe font metric) 파일과 외곽선(outline) 파일로 배포된다. 외곽선 파일은 보통 `pfb` (printer font binary) 또는 `pfa` (printer font ascii)파일이다. 외곽선 파일은 형태의 정보(glyph)를 갖는 반면, 메트릭(metric) 파일은 크기의 정보(metric)를 갖고 있다.

다른 플랫폼에 쓰이는 타입1 글꼴은 다른 포맷으로 배포될 수도 있다. 예를 들어 윈도즈에 쓰이는 포스트스크립트 글꼴은 종종 메트릭 파일로 다른 포맷(`pfm`)을 쓴다.

2.1.3 타입3 글꼴

이 글꼴도 타입1 글꼴과 비슷하게 `afm` 글꼴 메트릭과 `pfa` 파일의 묶음이라는 식으로 배포된다. 타입3 글꼴도 포스트스크립트 표준에 의해 지원되기는 하지만, X는 이 글꼴을 지원하지 않고 있다. 따라서 이 글꼴은 잘 쓰이지 않는다.

2.1.4 트루타입 글꼴

트루 타입 글꼴은 Apple에 의해 개발되었다. Apple사는 Microsoft가 이 포맷을 사용할 수 있도록 허용했으며, Adobe의 글꼴 시장 장악에 대해 도전하는데 성공했다. 트루 타입 글꼴은 크기와 형태의 정보를 하나의 파일(대개 `ttf`라는 확장자를 갖는다)에 저장한다. 최근 X에서 트루 타입 글꼴을 사용할 수 있도록 하는 글꼴 서버가 개발되었으며, 포스트스크립트와 고스트스크립트는 얼마전부터 트루타입 글꼴을 지원해 왔다. 이런 까닭으로, 트루타입 글꼴이 리눅스에서 점점 인기를 얻어가고 있다.

2.1.5 타입42 글꼴

타입42 글꼴은 포스트스크립트 해석기(interpreter)가 번역할 수 있도록 헤더(header)를 추가한 트루타입 글꼴이다. 고스트스크립트나 삼바와 같은 대부분의 응용프로그램들은 트루타입 글꼴을 사용자가 신경쓰지 않도록 (transparently) 처리하지만, 포스트스크립트 프린터를 가지고 있을 경우에는 타입42 글꼴 파일을 명시적으로 만들어야 하는 경우가 있을 수도 있다.

2.1.6 타입1 대 트루타입 – 비교

타입1 글꼴의 지지자와 트루타입 글꼴의 지지자 사이의 오랜 논쟁에도 불구하고, 이 두 글꼴은 많은 공통점을 가지고 있다. 둘 다 크기 변환이 가능한 외곽선 글꼴이다. 타입1 글꼴은 형태 정보(glyphs)를 위해 2차 곡선이 아니라 3차 곡선을 사용한다. 이것은 이론적으로는 트루타입 글꼴보다 더 많은 곡선을 표현할 수 있기 때문에 최소한 약간의 장점은 된다. 하지만 실제로는 거의 차이가 없다.

트루타입 글꼴은 hinting(역자주:비트맵으로 더 잘 표현되도록 글자의 외곽선을 조금씩 변경시키는 조작)을 더 잘 지원한다는 명백한 장점을 갖고 있다. (타입1 글꼴도 hinting 기능을 지원하기는 하지만 트루타입 글꼴 만큼 넓은 범위의 지원은 아니다.) 하지만, 이것은 스크린처럼 낮은 해상도를 갖는 장치에서나 문제가 된다. (600dpi 프린터에서는 작은 글꼴에서조차 hinting의 질에 따른 차이를 분간할 수 없다.) 잘 hint된(well hinted) 트루타입 글꼴이 드물다는 사실도, hinting이라는 이 명백한 장점을 의심스럽게 한다. 대부분의 소규모(small time) 디자이너들에게는 hinting 기능을 지원하는 소프트웨어 꾸러미란 너무 비싸다. 때문에 Monotype같은 소수의 대규모 글꼴업체(foundry)만이 잘 hint된 글꼴을 제공한다.

결론적으로, 트루타입 글꼴과 타입1 글꼴 사이의 주된 차이는 이용가능성 (availability)과 응용프로그램에서의 지원에 달려 있다. 트루타입 글꼴이 윈도즈에서 널리 쓰임에 따라, 웹 페이지들이 특정한 트루타입 글꼴이 이용가능하다는 것을 전제로 하고 디자인되게 되었다. 또한 트루타입 글꼴이 윈도즈 응용프로그램에 함께 실려 있기 때문에, 많은 사용자들이 대량의 트루타입 글꼴을 가지고 있다. 하지만 리눅스의 경우, 대부분의 응용 프로그램들이 타입1 글꼴을 지원하지만, 트루타입 글꼴에 대한 지원은 수준이 떨어진다. 더욱이 대부분의 대규모 글꼴 제작사들은 여전히 그들의 글꼴 대부분을 타입1 포맷으로 출하하고 있다. 예를 들어 Adobe는 트루타입 글꼴을 거의 내놓지 않는다. 사용자들에게 주는 필자의 충고는 무엇이 되었건 당신의 응용 프로그램에서 작동하는 글꼴을 쓰라는 것이다. 그리고 가능하면 한 포맷에서 다른 포맷으로 변환하는 것을 피하도록 노력하라는 것이다. (포맷 변환은 언제나 손실을 동반하기 때문에.)

2.1.7 메타글꼴(metafont)

메타글꼴는 Donald E. Knuth에 의해 TeX이라는 조판 시스템의 일부로 개발되었다. 메타글꼴는 포스트스크립트와 마찬가지로 그래픽 프로그래밍용 언어로서 글꼴에 그치지 않고 더 넓은 응용범위를 갖고 있다. 메타글꼴는 매우 바람직한 특성들을 보여주는데, 우아할 정도로 부드럽게 크기를 바꿀 수 있다는 것이 그 중요한 특징 중의 하나이다. 메타글꼴의 일종인 Computer Modern은 20 포인트와 10 포인트에서 서로 다른 모양을 갖고 있다. 모양이 크기에 따라 달라지는 것은 작은 글꼴가 큰 폰트에 비해 비교적 더 넓은 폭을 갖는 편이 바람직하기 때문이다. (이렇게 하면 큰 글꼴는 더 품위있어 보이고, 작은 폰트는 더 읽기 쉬워진다.)

메타글꼴는 흔히 `mf`라는 확장자를 갖는다. 메타폰트는 장치의존적인 비트맵 글꼴로 번역(render)되는데, 이 번역과정(rendering)이 느리기 때문에 훌륭한 특성을 가짐에도 불구하고 메타글꼴는 WISIWYG 출판에는 그다지 적합하지 못하다.

2.2 글꼴 족(families of typefaces)

서체는 대개 몇 가지 변형꼴들의 집합이라는 형태로 입수하게 된다. 예를 들어 대부분의 글꼴는 굵은 꼴(bold)과 기울인 꼴(italic), 굵은 기울인 꼴 (bold-italic)을 기본 꼴과 함께 갖는다. 어떤 글꼴는 작은 대문자나 약간 굵은 꼴(demibold)의 변형꼴을 갖기도 한다. 이렇게 하나의 글꼴와 그 변형꼴들로 이루어진 글꼴의 집합을 글꼴의 *족(family)*라고 부른다. 예를 들어 Garamond 글꼴 족은 Garamond, Garamond-italic, Garamond-bold, Garamond bold-italic, Garamond demi-bold, Garamond demi-bold-italic 으로 이루어져 있다. Adobe expert Garamond 글꼴에는 Garamond small caps, Garamond titling capitals까지 추가되어 있다.

제 3 절 Fonts 102 – 인쇄(typography)

이 절에서는 인쇄의 몇 가지 기본들에 대해 이야기 한다. 이 내용이 꼭 필요한 것은 아니지만, 글꼴을 좋아하는 사람이라면 흥미를 느낄 수 있을 것이다.

3.1 서체의 분류

3.1.1 고정폭 대 가변폭(fixed versus variable width)

글꼴에는 여러 가지 분류법이 있다. 첫번째로, **고정폭** 글꼴과 가변폭 글꼴로 나눌 수 있다. 고정폭 글꼴은 모든 글자들이 똑같은 폭을 갖기 때문에 타자기로 찍어낸 것 처럼 보인다. 이런 성질은 텍스트 편집기나 컴퓨터의 콘솔 같은 환경에서는 바람직하지만, 긴 글로 이루어진 본문에서는 좋은 것이 아니다. 고정폭 글꼴 가운데 가장 유명한 것이 쿠리어(courier) 글꼴이다. 다른 한 종류는 가변폭 글꼴이다. 고정폭 글꼴도 쓸모 있지만(예를 들어, 이 문서에서 예로 드는 모든 쉘 명령어는 고정폭 글꼴로 나타낸다.), 여러분이 쓰는 대부분의 글꼴은 가변폭이다.

3.1.2 세리프, 넣을 것인가, 넣지 않을 것인가?

세리프(serif)란 글자 끄트머리에 붙이는 조그만 갈고리 같은 것이다. 예를 들어 타임즈-로만 같은 글꼴에서는 i 자의 아랫쪽과 위쪽에 쑥 튀어나온 갈고리가 있다. 보통 세리프가 있는 글꼴이 없는 글꼴에 비해 더 읽기 쉽다고 한다. 세리프 글꼴에도 여러 가지 종류가 있다.

산세리프(sans serif=세리프 없는) 글꼴에는 이런 조그만 갈고리가 없어서, 비교적 뻣뻣해 보인다. 산세리프 글꼴은 책의 본문 같은 곳에는 잘 쓰이지 않지만, 간단히 훑어 보게 되는 문서(웹 페이지, 카탈로그, 판매용 팸플릿 등)에 적합하다. 산세리프 글꼴은 컴퓨터 스크린에 표시되는 글꼴, 특히 작은 크기의 글꼴로도 이용된다. 글꼴에 세세한 부분이 없기 때문에, 산세리프 글꼴이 더 분명히 표시되기 때문이다. 예를 들어 Microsoft는 Verdana 글꼴을 스크린 상에서 매우 작은 크기로도 읽을 수 있게 하도록 요구하고 있다.

주목할 만한 산세리프 글꼴에는 Lucida sans, MS Comic Sans, Verdana, Myriad, Avant Garde, Arial, Century Gothic, Helvetica 등이 있다. 한편 인쇄업자들은 Helvetica가 해롭다고 생각한다. Helvetica는 다소 남용되고 있으며, 인쇄업자들이 쓴 많은 책들이 사용자들에게 Helvetica로부터 거리를 두어달라고 요청하고 있다.

3.1.3 묵은 것과 새 것 – 세리프 글꼴의 여러 형태

올드스타일(Old Style) 올드스타일 글꼴은 15세기 후반까지 거슬러 올라가는 아주 전통적인 스타일에 바탕을 두고 있다. 이런 글꼴은 보수적인 디자인을 따르는 경향이 있고, 매우 읽기 좋다. 이 글꼴들은 긴 문서를 쓰기에 적합하다. '올드스타일'이라는 이름은 글꼴이 디자인된 때가 아니라, 글꼴의 스타일을 말하는 것이다. 예를 들어 Goudy Old Style 같은 전형적인 올드스타일 글꼴이 있는데, 이 글꼴은 20세기에 디자인된 것이다. 올드스타일 글꼴들은 다음과 같은 특색을 갖고 있다.

- 잘 정의된 아름다운 세리프.
- 대각선 방향의 강조. 펜으로 수직선에서 시계반대 방향 45도의 선은 굵게, 시계 방향 45도의 선은 가늘게 쓴 글꼴을 상상해 보라. 올드스타일 글꼴은 이런 모양을 갖는 경우가 많다.
- 읽기 쉬움. 올드스타일 글꼴은 언제나 아주 읽기 좋다.
- 모호함, 대조의 결여. 올드스타일 글꼴은 굵은 선과 가는 선을 갖고 있지만 그 굵기의 대비는 모호하고 엄격하지 않다.

주요한 올드스타일 글꼴로는 Garamond, Goudy Old Style, Jenson, Caslon 등이 있다. (마지막 것은 이론의 여지가 있다. 트래디셔널로 보는 이들도 있다.)

모던(Moderns (or didone)) 모던 글꼴은 올드스타일 글꼴과 반대다. 모던 글꼴은 대개 올드스타일 글꼴에 비해 더 많은 문자와 더 많은 속성을 갖고 있으며, 긴 문장을 입력하기 보다는 문서에 문자를 추가하는 쪽에 적합하다. 하지만, 흑백으로 갈리는 것은 아니어서, computer modern이나 Monotype modern, New Century Shoolbook 처럼 아주 읽기 좋은 모던 글꼴들도 있다. (이 글꼴들의 경우, 더 읽기 좋게 하기 위해 굵은 획과 가는 획 사이의 대비가 더 부드럽게 되어있다.) 모던 글꼴은 19세기 이후에 유행한 디자인에 바탕하고 있으며, 다음과 같은 특징들을 갖고 있다.

- 세리프가 덜하고, 가는 수평선 만으로 처리한 경우도 많다.
- 수직선의 강조. 수직선은 굵고, 수평선은 가늘다.
- 가는 획과 굵은 획 사이의 대비가 엄격한 경우가 많다.
- 모던 글꼴은 가는 획과 굵은 획 사이의 대비가 강해서 올드스타일 글꼴에 비해 읽기에는 좋지 않다.

Bodoni가 가장 유명한 모던 글꼴이다. 다른 모던 글꼴로는 computer modern과 computer modern의 바탕이 된 Monotype modern이 있다.

트래디셔널(Transitional) 트래디셔널 글꼴은 모던과 올드스타일 글꼴의 중간쯤에 해당한다. 많은 트래디셔널 글꼴이 올드스타일 글꼴에 버금가게 읽기 좋지만, 올드스타일 보다는 후대의 디자인에 바탕하고 있다. 모던 글꼴로 향한 움직임을 보여주기도 하지만, 트래디셔널 글꼴들은 모던 글꼴에 비해서는 훨씬 더 모호하다. 트래디셔널 글꼴의 예로는 Times Roman, Utopia, Bulmer, Baskerville 등이 있다. 이 중에서도 Times는 올드스타일에 가까운 반면, Bulmer는 모던처럼 보인다.

슬랩세리프(Slab Serif) 슬랩세리프 글꼴은 올드스타일의 부드러운 갈고리꼴이나 모던의 일부에서 보이는 가는 선에 비해, 두껍고 마치 벽돌 같은 세리프를 갖기 때문에 그런 이름을 갖게 되었다. 슬랩 세리프 글꼴은 견고해 보이고 일반적으로 매우 읽기 좋다. 슬랩 세리프 글꼴에는 Nile이나 Egyptienne 처럼 이집트와 관련된 이름이 많다. (이 글꼴들은 실제로는 이집트와 아무 관계가 없다.) 여기 속하는 글꼴들은 복사된 문서나 신문에 인쇄되는 문서 처럼 질의 저하에 의해 문제가 생길 수 있는 문서를 읽기 쉽게 하는데 매우 매우 훌륭하다. 이 글꼴은 정말로 견고하다. 유명한 슬랩 세리프 글꼴로는 여러 가지 타자기 글꼴과 Clarendon, Memphis, Egyptienne가 있다. 슬랩 세리프 글꼴에는 고정폭인 것이 많다. 거꾸로 대부분의 (거의 모든) 고정폭 글꼴은 슬랩 세리프에 속한다.

3.1.4 산세리프 혁명(The Sans Serif Revolution)

놀랍게도 산세리프 글꼴의 등장은 상당히 최근의 일이다. 최초의 잘 알려진 산세리프 글꼴은 19세기에서 20세기 초에 도안되었다. 초기의 도안에는 Futura와 Grotesque, Gill Sans가 있다. 이 글꼴들은 산세리프 글꼴 가운데 '지오메트릭', '그로테스크', '휴머니스트' 부류를 각각 대표한다.

그로테스크(grotesque) 그로테스크라는 이름은 처음에 사람들이 상대적으로 뻣뻣한 그 디자인에 놀랐기 때문에 붙여졌다. 그로테스크는 세리프가 없고 보다 단순 명료한 디자인 때문에 전혀 장식이 없어보인다. 이런 '충격적인(in your face)' 외관 때문에 그로테스크는 헤드라인에 알맞고, 보다 읽기 좋게 한 변형꼴들은 본문이 큰 역할을 하지 않는 만화책이나 판매용 팸플릿 등에 좋다. 그로테스크는 맞수격인 지오메트릭 글꼴에 비해 예술적으로 보이지는 않는다. 지오메트릭 글꼴에 비해, 그로테스크는 폭에 더 많은 변화가 있고, 더 많은 획을 가지며, 원호 등을 쓰지 않기 때문에 보다 사각꼴이다. 그로테스크는 대문자 G와 소문자 a를 지오메트릭과 달리 쓴다. 그로테스크는 최소한의 디자인을 추구하기는 하지만, 무지막지하게 전위적인 지오메트릭 처럼 극단 까지 나아가지는 않는다.

주목할 만한 그로테스크 글꼴로는 남용되고 있는 Helvetica를 비롯해, Grotesque, Arial, Franklin Gothic, Univers가 있다.

지오메트릭(geometric) Futura 글꼴은 *형태는 기능을 따른다(form follows function)*라는 선언으로부터 나왔다. 지오메트릭 부류의 글꼴은 엄격하게 최소한의 디자인을 추구하는 외관을 갖는다. 경중이 없는 일정한 선폭이 중요한 특징 중의 하나이다. 굵은 그로테스크와 휴머니스트 글꼴이 때때로 눈에 띄는 선의 변화를 보이는데 비해, 지오메트릭 글꼴에서는 이런 변화가 극히 드물다. 디자인 상에서 확실하게 지켜지는 최소주의도 주목할 만 하다. 글자들은 거의 언제나 곧은 수평선과 수직선으로 이루어지고, 호는 마치 컴퍼스로 그려진 것 처럼 보일 만큼 원형이다. 글자들은 최소한의 획수를 갖는다. 나중에 근대 예술을 폭풍처럼 휩쓴 최소주의 철학을 채용하고 있다는 점에서, 이런 특징은 지오메트릭 글꼴에 현대적인 면모를 준다. 글꼴이 지오메트릭 부류라는 것을 말해주는 표시는 대문자 G인데, 이 글자는 최소주의에 입각해 길쭉한 원호 하나와 수평선 하나라는 두 획의 결합으로 구성되어 있다. 눈에 띄는 다른 하나의 글자는 소문자 a로서, 이 글자도 수평선과 원이라는 단순한 두 획으로 되어 있다. (또다른 a 자는 더 복잡해서 쓰이지 않는다.) 주목할 만한 지오메트릭 글꼴로는 Avant Garde, Futura, Century Gothic이 있다.

휴머니스트(humanist) 이름이 말해주듯이, 휴머니스트 글꼴은 외관에서 보다 인간적이 되자는 목적을 가지고 도안되었다. 많은 면에서 여기 속하는 글꼴은 지오메트릭이나 그로테스크 보다 셰리프 글꼴을 닮았다. 이 글꼴은 '펜으로 쓰여진' 모양을 가졌다고 일컬어진다. 이 글꼴은 선의 굵기에 미묘한 변화를 갖는 경향이 있는데, 특히 굵은 글꼴에서 눈에 띈다. 곡선의 모양은 지오메트릭에 비해 훨씬 덜 엄밀하다. 여기 속하는 글꼴들은 "double story"의 소문자 g에 의해 다른 글꼴들과 구분되는데, 올드스타일의 셰리프 글꼴에서 쓰이는 g와 같은 모양을 하고 있다. 휴머니스트 서체는 올드스타일 글꼴과 비교적 호환 가능하기 때문에, 꼴사나운 문서를 만드는 일 없이 가장 손쉽게 쓸 수 있다.

3.1.5 호환성있는 서체(Compatible Typefaces)

서체를 그룹으로 묶는 것은 쉬운 일이 아니고, 따라서 한 페이지에 너무 많은 글꼴을 사용하는 것은 피하는 편이 이롭다. 두 가지 서체를 논리적으로 선택하자면 셰리프와 산셰리프다. *Monotype's Typography 101 page* `<http://www.monotype.com/newmedia/type101_ex.htm>` 에는 어울리는 서체들을 범주로 묶어 알려주고 있다. 이 페이지에서는 모던과 지오메트릭 서체가 좋은 짝을 이루고, 올드스타일과 휴머니스트도 잘 어울린다고 결론을 내리고 있다. 트래디셔널도 휴머니스트와 잘 어울리는 한 쌍이다. 슬랩셰리프는 그로테스크와 짝을 이루고, 슬랩셰리프의 변형꼴 중에는 지오메트릭이나 휴머니스트와 어울리는 것도 있다고 한다.

이 홈페이지를 읽고 나면, 이 홈페이지의 생각이 본질적으로 좀더 보수적인 셰리프는 좀더 온건한 산셰리프와 어울리고, 보다 거친 모던 셰리프는 전위적으로 보이는 지오메트릭과 쌍을 이룬다는 것이라는 느낌을 받게 된다.

3.2 모은자와 작은 대문자 글꼴, 그리고 전문가 글꼴

3.2.1 모은자(Ligatures)

글꼴의 간격을 적절히 맞추는 것은 많은 문제를 제기한다. 예를 들어 "fi"라고 제대로 글자를 조판하기 위해서는 i가 f에 아주 가깝게 위치해야 한다. 문제는 이렇게 할 경우 i의 윗부분의 점이 f와 부딪히고, i 머리부분의 셰리프도 f의 수평획과 부딪힌다는 것이다. 이런 문제를 해결하기 위해서 글꼴모음에는 모은자가 포함되어 있다. 예를 들어 "fi"라는 모은자는 "fi"라는 두 글자로 된 문자열을 대신해서 쓸 수 있는 한 글자다. 대부분의 글꼴에 fi와 fl의 모은자가 들어 있다. 나중에 거론할 전문가(expert) 글꼴들은 ffl, ffi와 윗점을 땐 i 자 등의 다른 모은자 까지를 포함하는 경우도 많다.

3.2.2 작은 대문자 글꼴(Small caps fonts)

작은 대문자 글꼴은 소문자 대신 축소된 대문자를 갖고 있다. 이런 글꼴은 강조할 필요가 있는 제목 등을 쓰는데 유용하다. (LaTeX에서 자주 사용된다.) 대개 작은 대문자로 제목을 쓸 경우, 각 낱말의 첫 글자는 큰 글꼴로 쓰고, 나머지는 작은 글꼴로 쓴다. ("title case"라고 한다.) 이렇게 하면, 모두 큰 글꼴로 썼을 때에 비해 훨씬 더 읽기 좋아지는 장점이 있다. (모두 큰 글꼴로 쓰는 것은 조판계에서는 커다란 죄악이다.)

3.2.3 전문가 글꼴(Expert fonts)

전문가 글꼴은 하나의 서체를 보완하기 위해 도안된 여러 가지 추가 글꼴로 이루어진다. 이 글꼴에는 모은 자, 서체와 함께 쓰기 위해 도안된 작은 장식 활자(mini-dingbat)과도 같은 오너먼트(ornament), 작은 대문자 글꼴, 멋을 부려 서예체로 쓴 붓글씨체(swash) 문자 등이 포함된다.

3.3 글꼴의 메트릭과 형태(Font Metrics and Shapes)

글꼴의 메트릭은 여러 가지 폭의 글꼴들 사이의 간격을 정의한다. 메트릭은 글꼴의 크기와 *커닝(kerning)* 정보를 포함하고 있다. 커닝 정보란 커닝 짝, 즉 서로 다른 간격을 주어야 하는 글자쌍들을 지정하는 정보이다. 예를 들어 "To"는 보통 커닝 짝에 속하는데, 간격이 바르게 주어지면(kerned), o의 일부분이 T의 아래에 있어야 하기 때문이다. LaTeX과 같은 조판 프로그램은 어디에서 한 줄을 끊고, 어디에서 다음 페이지로 넘어갈까를 판단하기 위해 커닝에 대한 정보를 알아야 한다. WYSIWYG 인쇄 프로그램에서도 사정은 마찬가지다.

메트릭에 더하여, 글꼴의 외곽선 혹은 형태가 있다. 획이나 액센트 표시 등, 글꼴의 형태를 이루는 구성요소는 "글리프(glyphs)"라고 불리운다.

제 4 절 X에서 글꼴을 사용할 수 있게 하기

X에 글꼴을 추가하는 방법은 여러 가지가 있다. XFree86은 글꼴을 찾아볼 여러 디렉터리와 *글꼴 서버*의 목록인 *글꼴 경로*를 갖고 있다. 글꼴 서버란 XFree86가 글꼴을 사용할 수 있도록 해주는 백그라운드 프로세스다. 글꼴 서버의 장점은 원격 디스플레이에도 글꼴을 보내 줄 수 있다는 점이다.

최근 `xfs`("X font server")가 트루타입 글꼴을 지원하고, 독립적인 (stand-alone) 프로그램으로 동작하도록 패치가 이루어졌다. 패치된 버젼은 레드햇과 레드햇에 기반한 배포판에 들어있으며, XFree86의 3.9.17에 포함되어 있다. (이것이 이 글을 쓰는 시점의 최신 버젼이다. 패치된 xfs는 XFree86 4.0에도 일부분으로 포함될 예정이다.) `xfs`는 사실 XFree86의 표준 글꼴 서버이며, 그 소스 코드는 XFree86의 소스 트리의 일부이다. 하지만, 최근의 배포판에는 XFree86과 독립적으로 동작하는 버젼이 들어있다. 트루타입을 지원하는 패치가 가해진 독립적인 X 글꼴 서버는 아마도 현재 가능한 최선의 글꼴 관리 방법일 것이다. (트루타입에 대한 지원은 `xfsft`라 불리는 글꼴 서버에 의해 이루어진다.) 이런 방식은 다음과 같은 장점을 갖는다.

- 타입1, 트루타입, 비트맵 등을 포함하는 여러 종류의 글꼴에 대한 지원.
- 원격 디스플레이에서도 글꼴을 사용할 수 있도록 한다.
- 글꼴 경로를 지극히 간단히 편집할 수 있게 한다. 설정 파일들을 편집하는 것이 아니라, `chkfontpath`라는 명령행 유틸리티를 이용하여 글꼴 경로의 편집이 가능하다. 이점은 사용자를 더 편하게 할 뿐아니라, 꾸러미를 만드는 쪽에서도 더 다양한 스크립트를 이용하여 더욱 안전하게 꾸러미를 만들 수 있도록 한다.

여러 배포판이 저마다 다른 설정을 가지고 있기 때문에, 한 가지 방식으로 모든 배포판에 대응할 수는 없다. 다음과 같은 세 가지 그룹으로 나누어 설명하자.

- 트루타입을 지원하는 패치가 된 독립적인 `xfs`가 들어있는 배포판 사용자. 이 그룹에는 레드햇 사용자와, 맨드레이크, 터보리눅스, 인디펜던스 등 레드햇 기반의 배포판 사용자가 포함된다. 이 그룹은 트루타입과 타입1 글꼴 양쪽을 `xfs`를 통해 설치하는 것이 가장 현명한 전략이다.
- 이 글을 쓰고 있는 시점에는 독립적인 `xfs`는 들어있지만, 트루타입 지원은 안되는 배포판도 있다. (XFree86은 3.9.17 버젼 부터 트루타입을 지원하고 있다는 점에 유의하라.) 데비안이 이런 경우에 속하는데, 이런 배포판의 사용자의 경우에는 타입1 글꼴을 설치하는 데에는 `xfs`를 쓰고, 트루타입 글꼴은 `xfstt`로 설치하는 것이 최선이다. 데비안 사용자는 *TrueType Fonts in Debian mini-HOWTO* `<http://www.dimensional.com/~bgiles/debian-tt.html>` 에서 트루타입 글꼴 설치에 대한 정보를 찾아볼 수 있다.
- `xfs`를 갖고 있지 않다면 타입1 글꼴이 들어있는 디렉터리를 XFree86의 글꼴 경로에 추가하고 xset을 이용함으로써 타입1 글꼴을 설치해야 한다. 트루타입은 `xfstt`로 설치해야 한다.

4.1 글꼴 경로(The font path)

XFree86은 *글꼴 경로*라 불리는 글꼴이 들어있는 디렉터리(혹은 서버들 – 여기에 대해서는 나중에 더 설명한다)의 목록을 검색함으로써 글꼴을 찾는다. 응용 프로그램이 어떤 글꼴을 요구하면, XFree86은 그 글꼴이 발견될 때 까지 글꼴 경로에 들어있는 디렉터리들을 하나 하나 검색한다. 따라서 글꼴을 사용할 수 있도록 하려면, 글꼴 경로를 지정해야 한다. 다음의 명령으로 글꼴 경로에 임의의 디렉터리를 추가할 수 있다.

```
xset fp+ directory
```

일단 이 명령을 수행한 다음에는 X 서버가 가능한 글꼴을 다시 조사하도록 다음의 명령을 내려야 한다.

```
xset fp rehash
```

이 명령이 자동적으로 수행되기를 원한다면, 자신의 `.xinitrc` 파일에 넣어두어야 한다. (혹은 `.Xclients`나 `.xsession` 파일에 넣어도 된다. 어느 파일에 넣는가는 X를 실행시키는 방식에 따라 다르다. 혼란을 피하기 위해서는 이 파일들을 심볼릭 링크시켜두도록 한다.) 이런 명령들이 자동적으로 실행되도록 하는 또다른 방법은 XF86Config 파일을 편집하는 것이다. 예를 들어 `/usr/share/fonts/myfonts` 디렉터리를 X가 시작될 때 글꼴 경로에 추가하려면 `XF86Config`를 다음과 같이 편집한다.

```
...
Section "Files"
...

FontPath /usr/share/fonts/myfonts
...
EndSection
...
```

XF86Config를 편집했을 때의 장점은 시스템 전체에 걸쳐 변화를 준다는 것이다.

4.2 타입1 글꼴의 설치

4.2.1 type1inst 실행하기

타입1 글꼴을 X에서 사용가능하게 하는 가장 쉬운 방법은 Type1inst 유틸리티의 도움을 받는 것이다. 이 유틸리티는 타입1 글꼴을 X에서 쓰기 위해 필요한 fonts.dir와 fonts.scale 파일들을 자동적으로 만들어주는 펄스크립트이다. 간단히 글꼴 디렉터리로 이동해서, type1int를 수행한다.

```
cd directory
type1inst
```

4.2.2 xfs 꾸러미를 가지고 있을 경우

이제 타입1 글꼴을 글꼴 경로에 추가해야 한다. 이미 독립적인 4.4 (xfs)가 수행되고 있다면, `xfs`의 설정 파일을 편집하면 된다. *레드햇 사용자라면 4.4.1 (chkfontpath) 를 사용하기만 하면 된다. 형식은 다음과 같다.* `chkfontpath --add directory`

이제 설치한 타입1 글꼴을 X에서 쓸 수 있게 되었다. 다음에는

```
xset fp rehash
```

를 수행하기만 하면, X가 새로운 글꼴을 찾을 수 있게 될 것이다.

4.2.3 xfs 꾸러미가 없을 경우

이 경우에는 앞에서 설명한 것과 같이, 새로운 글꼴이 들어있는 디렉터리를 글꼴 경로에 추가해 주어야 한다.

4.3 트루타입 글꼴(True Type Fonts)

트루타입 글꼴을 추가하는 것은 타입1보다 어렵다. 트루타입 글꼴을 지원하는 글꼴 서버를 갖고 있어야 하기 때문이다. 이런 기능을 제공하는 것에는 xfstt와 xfs의 두 가지 글꼴 서버가 있다.

`xfstt`는 트루타입 글꼴 서버이다. 이쪽이 설정이 간편하고 상당히 유용함에도 불구하고, `xfs`가 더 인기를 얻어가는 듯하다. `xfstt`에 비해 볼 때, `xfs`의 주된 장점은 타입1과 트루타입 양쪽을 모두 지원한다는 점이다.

4.3.1 xfstt

xfstt를 사용하려면 다운받아서 설치하기만 하면 된다. rpm 기반의 배포판이라면 http://independence.seul.org/ 에 잘 패키징된 버젼이 있다. xfstt를 설치한 다음에는 다음과 같이 해 주어야 한다.

1. 글꼴을 적절한 디렉터리에 설치한다. (꾸러미에 딸려오는 문서를 읽자.)
2. 그 디렉터리로 가서 `xfstt --sync`를 실행한다. 이 명령은 xfstt에게 글꼴을 찾아서 `fonts.dir` 파일을 만들라고 하는 것이다.
3. 이제 `unix/:7100`를 글꼴 경로에 더한다.

이제 트루타입 글꼴이 GIMP나 넷스케이프같은 응용 프로그램에 표시되고 사용될 수 있어야 한다. xfstt가 시스템이 시작될 때마다 실행되도록 설정하고 싶을 수도 있다. 먼저 시작(startup) 파일이 꾸러미에 포함되어 있는지 확인하자. (RPM을 쓰고 있다면 `rpm -ql xfstt |grep init`를 이용할 수 있다. 그리고 나서 `/etc/rc.d/init.d/xfstt` 등의 이름을 가진 파일을 살펴보자.) 만약 시작 스크립트가 없다면, `/etc/rc.local` 디렉터리에 다음의 두 줄(시스템에 따라 조금 다를 수도 있다)을 가진 스크립트를 만들어 넣으면 된다.

```
/usr/X11R6/bin/xfstt --sync
/usr/X11R6/bin/xfstt &
```

4.4 xfs

근래 나오는 리눅스 배포판 중에는 독립된 프로그램으로 작동하도록 설정된 X 글꼴 서버 xfs가 들어 있는 것들이 있다. 특히, 레드햇 및 레드햇 기반의 배포판들이 이렇게 트루타입을 지원하도록 컴파일된 모듈방식의 xfs를 사용하고 있다. 데비안에도 xfs가 들어있지만, 데비안의 xfs는 트루타입을 지원하도록 되어 있지 않다.

xfs를 독립적인 서버로 돌리는 것은 여러 가지 이점을 갖고 있다. 특히 트루타입을 지원하도록 컴파일되어 있다면 장점은 더욱 두드러진다. 주된 장점은 글꼴 서버가 더이상 X 서버에 결합되어 있지 않기 때문에, 원격 디스플레이로 글꼴을 제공할 수 있다는 점이다. 아울러 글꼴 경로를 변경하는 것도 훨씬 쉽게 되었다.

4.4.1 xfs 경로

xfs는 글꼴 서버로서 자체의 글꼴 경로를 갖는다. 이 패스가 X에서의 글꼴 사용이라는 전체 그림 상의 어느 부분에 해당되는지 궁금할 것이다. xfs는 다음과 같은 방식으로 작동한다. XFree86의 글꼴 경로에 unix/:port를 추가함으로써 xfs 글꼴 서버를 XFree86의 글꼴 경로에 넣는다. 이렇게 하면 자동적으로 xfs의 글꼴 경로 안에 있는 모든 글꼴을 XFree86에서 사용할 수 있게 된다.

xfs의 글꼴 경로는 xfs의 설정 파일에 의해 결정되는데, 이 설정 파일은 레드햇에서는 /etc/X11/fs/config이고, 데비안에서는 /etc/X11/xfs/config이다. 레드햇 사용자라면 이 파일을 직접 편집할 필요없이 chkfontpath라는 유틸리티를 사용할 수 있다. 사용법은 간단하다.

```
chkfontpath --add directory
```

다른 배포판의 사용자는 다음과 같이 설정 파일을 편집할 수 있다.

```
catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
...
/usr/share/fonts/my_new_fonts/,
...
/usr/share/fonts/some_other_directory
# in 12 points, decipoints
default-point-size = 120
...
```

위의 예는 /usr/share/fonts/my_new_fonts/ 디렉터리를 xfs의 글꼴 경로에 더하는 것이다. *디렉터리 목록의 마지막 줄 끝에 쉼표가 없다는 것에 주의하여야 한다.* 글꼴 경로를 이렇게 바꾼 것이 효과를 내기 위해서는 xfs를 다시 시작시켜야 한다. xfs를 다시 시작시킨 다음에 X를 다시 시작시키는 것도 좋은 생각이다.

4.4.2 xfs에 글꼴을 설치하기

xfs에서 글꼴을 사용하기 위해서는 다음과 같은 단계를 거쳐야 한다.

- xfs를 설치하지 않았다면, 설치하라.

- 새 글꼴을 원하는 디렉터리에 넣는다.
- 타입1 글꼴을 설치한다면 `type1inst`를 글꼴 디렉터리 안에서 실행하여 새 디렉터리를 서버가 사용할 수 있도록 준비한다.
- 트루타입 글꼴을 설치한다면,

```
ttmkfdir -o fonts.scale
mkfontdir
```

  를 새 글꼴이 들어있는 디렉터리에서 실행함으로써, 새 디렉터리를 서버가 사용할 수 있도록 준비한다. (*모든 배포판에 트루타입을 지원하는 xfs가 있지는 않다는 점을 기억하자!*) `ttmkfdir`는 *freetype* 꾸러미의 일부이다.

- 이제 새로운 디렉터리를 `xfs`의 검색 경로에 추가할 수 있다. 레드햇 기반의 배포판을 쓰고 있다면 `chkfontpath` 유틸리티를 사용해서 이 작업을 할 수 있다. 다른 배포판의 경우에는 `xfs` 설정 파일을 편집해서 추가한다.
- `xfs`가 이미 설치되어 있다면, `xfs`가 어느 포트에서 작동하고 있는지 알아야 한다. 다음과 같이 한다.

```
ps ax|grep xfs
```

- 다음에는 XFree86의 글꼴 경로를 확인한다.

```
xset -q
```

- 글꼴 경로가 `unix:/`*port_number* 비슷한 내용을 포함하고 있다면 xfs는 제대로 설정된 것이다. 여기서 *port_number*란 서버가 돌아가고 있는 포트다. 만약 제대로 설정 되어 있지 않다면, 다음의 방법으로 이 부분을 XFree86의 글꼴 경로에 더해 주어야 한다.

```
xset fp+ unix/:port_number
xset fp rehash
```

  앞서 설명한 바와 같이 자신의 `.xinitrc`를 편집함으로써 위의 글꼴 경로를 영구적으로 XFree86의 글꼴 경로에 포함시킬 수도 있다. 시스템 전체에 걸쳐 XFree86의 글꼴 경로에 포함시키려면, 시스템의 XF86Config 파일의 Files 절에 `FontPath "unix:/port_number"`라는 줄을 추가한다. (시스템의 XF86Config는 `/etc/X11/XF86Config`나 `/etc/XF86Config`, `/usr/X11R6/lib/X11/XF86Config` 같은 파일이다.) 다음은 하나의 예이다.

```
...
Section "Files"
...

FontPath "unix/:-1"
...
EndSection
...
```

- `xfs`가 제대로 설치되었다면, 다음과 같이 `xfs`를 다시 시작하도록 한다.

```
/etc/rc.d/init.d/xfs restart
```

- xfs를 다시 시작한 다음에는 X 세션을 다시 시작하는 것이 좋다.

제 5 절 고스트스크립트에서 글꼴을 사용할 수 있게 하기

고스트스크립트에서 글꼴을 사용할 수 있게 하려면, 주어진 글꼴에 해당하는 파일이 어디에 있는지 고스트스크립트에게 알려주기만 하면 된다. 편집해야 할 파일은 /usr/share/ghostscript/version/Fontmap이다. 형식은 매우 간단해서, 한 번 읽어보면 거의 뻔하다.

5.1 Type1

타입1 글꼴을 추가하는 것은 간단하다. 글꼴이 있는 디렉터리에서 type1inst를 실행한다. type1inst는 Fontmap이라는 파일을 만들어 낸다. 이 파일을 고스트스크립트의 Fontmap 파일 뒤에 이어붙인다.

5.2 True Type

트루타입 글꼴을 추가하려면 약간 더 재주를 부려야 한다. 트루타입 글꼴의 이름을 알아내야 하기 때문이다. 한 가지 방법은 (무식한 짓이다, 슬프게도.) ttf2pt1를 써서 트루타입을 타입1으로 변환시킨 다음 afm에서 글꼴 이름을 얻어내는 방법이다. (더 효과적인 방법이 있을 테지만, 이 방법도 통한다. 꼴사납긴 하지만.) 다음과 같이 한다.

```
ttf2pt1 -A fontname - 2 > /dev/null  |grep FontName
```

다음에는 이 목록을 고스트스크립트의 Fontmap 파일에 아래의 예와 같이 정확한 형식으로 추가한다.

```
some-font       (/usr/share/fonts/subdirectory/somefont.pbf);
```

이런 방식으로 하면 되지만, 수백개의 글꼴을 다룬다고 해보자. 이런 작업은 짤막한 펄스크립트가 필요한 일이다.

```
#!/usr/bin/perl
# ttfontmap -- generate fontmap file for TrueType fonts
my $directory=shift || print STDERR "Usage: ttfontmap {directory}\n";

$directory=~s/\/$//;

for my $fontname ( glob ( "$directory/*.ttf" ) )
{
    open ( R, "sh -c \"ttf2pt1 -A $fontname - 2>/dev/null\" |" );
    while ( <R> )
    {
        if ( $_ =~ /^FontName/ )
        {
            s/^FontName\s*//;
            chomp;
```

```
            print "/" .  $_ . "    ($fontname);\n" ;
        }
    }
    close R;
}
```

누구나 이 스크립트 <http://pegasus.rutgers.edu/~elflord/font_howto/ttfontmap> *를 다운받을 수 있다.*

이 스크립트를 만들려면, 위의 부분을 잘라서 ttfontmap이라는 파일로 만들고, 이 파일을 /usr/bin 등등 자신의 PATH에 들어있는 디렉터리에 두기만 하면 된다. 사용 방법은 다음과 같다.

```
        ttfontmap directory > output_file
```

directory는 글꼴이 들어있는 디렉터리다. 위와 같이 하면 고스트스크립트의 fontmap 파일에 이어 붙일 수 있는 output_file 파일이 생긴다. 주의: 그저 다음과 같이 해도 된다고 생각할 수도 있다.

```
        ttfontmap directory >> /usr/share/ghostscript/version/Fontmap
```

하지만, 이렇게 하지 않기를 권한다. (실수로 “>>“ 대신 “>”라고 하면 무슨 일이 일어날까?)

5.3 글꼴 미리보기를 위해 고스트스크립트를 쓰기

글꼴을 고스트스크립트에서 쓸 수 있도록 하고 나면, 그 글꼴의 미리보기가 가능하다. 먼저 설치된 고스트스크립트 꾸러미 안에 있는 prfont.ps에 대해 고스트스크립트를 실행시키고, 고스트스크립트 글꼴에 다음과 같이 입력한다.

```
        /Fontname DoFont
```

(FontName은 미리보기를 하고 싶은 글꼴이 고스트스크립트에서 갖는 이름이다.) gs를 기동시키는 방법은 이밖에도 여러 가지가 있다. 예를 들면 gv처럼 좀 더 멋진 포스트스크립트 뷰어를 써서 볼 수 있도록 포스트스크립트 파일을 만들기를 원한다면, 다음과 같이 한다.

```
        gs -sDEVICE=pswrite -sOutputFile=somefile.ps  prfont.ps
```

이렇게 한 다음에는 출력된 파일을 인쇄할 수도 있다.

제 6 절 트루타입에서 타입1으로의 변환

6.1 왜?

혹은 ”왜 안하나?”라는 쪽이 올바른 질문일 지도 모르겠다. 평범한 리눅스 사용자라면 윈도우즈로부터 건너온 경험을 갖고 있을 것이고, 아마도 엄청난 수의 트루타입 글꼴을 모아두고 있을 것이다. 이 글꼴들 가운데에는 MS 워드나 코렐 사의 제품에 들어있는 것들 처럼 상당히 훌륭한 것들이 많이 있다. 하지만 리눅스 응용 프로그램들 중에는 스타 오피스나 LaTeX와 같이 트루타입 글꼴은 지원하지 않지만 타입1 글꼴은 지

원하는 것들이 있다. *갱신: 스타 오피스는 트루타입을 처리할 수 있는 듯하다. 하지만 글쓴이는 아직 세밀한 부분을 해결하려고 노력하는 중이다. 어떻든, 지금으로서는 그 해결책이 상당히 심한 해킹까지 포함하고 있다.* 일부 프로그램이 트루타입을 지원하지 않는다는 것은 애석한 일이다. 고스트스크립트가 트루타입을 지원하고, 트루타입 글꼴 서버가 있는 등 리눅스에는 트루타입을 처리하기 위해 필요한 기본구조가 갖추어져 있기 때문이다.

6.2 어떻게?

트루타입을 타입1 글꼴로 변환하려면, *http://quadrant.netspace.net.au/ttf2pt1/* `<http://quadrant.netspace.net.au/ttf2pt1/>` 로 가서 ttf2pt1을 다운받는다. 트루타입을 타입1 글꼴로 변환하려면 다음과 같은 사용법을 따른다.

```
ttf2pt1 -b file.ttf name
```

여기서 `name`은 새로운 타입1 글꼴에 해당하는 파일의 이름이므로 자기 마음대로 해도 된다. 그래도 eg `ttf2pt1 -b foo.ttf foo` 처럼, ttf 파일과 같이 만드는 편이 좋은 생각이다.

자, 위와 같이 하면 하나의 글꼴은 잘 처리할 수 있다. 글꼴을 많이 갖고 있다면, 더 영리한 방법이 필요하다. 다음과 같은 반복문을 하나 사용하면 된다.

```
for X in *.ttf; do ttf2pt1 -b $X ${X%%.ttf}; done
```

아니면, *ttfutils* `<http://www.moisty.org/~brion/linux/ttf.html>` 꾸러미를 다운받아서 변환에 `ttf2type1`을 쓸 수도 있다.

```
ttf2type1 *.ttf
```

제 7 절 WYSIWYG 출판과 글꼴

7.1 소개와 개관

리눅스 상에서 WISIWYG 인쇄를 위해 글꼴을 설치하는 것은 상대적으로 복잡한 과제다. 이 작업은 보통 세 단계로 이루어진다.

- X 서버에서 글꼴을 사용할 수 있도록 만드는 단계
- 고스트스크립트에서 글꼴을 사용할 수 있도록 하는 단계
- 특정 응용 프로그램에서 글꼴을 사용할 수 있도록 하는 단계

이렇게 복잡해지는 것는 *글꼴 인쇄 시스템*(고스트스크립트)가 *화면 글꼴 시스템*과 연결되어 있지 않기 때문이다. 이를테면, 리눅스에서는 오른손이 하는 일을 왼손이 모르는 것이다. 이 문제를 해결하는 것은 간단한 일이 아니다. 인쇄용 글꼴과 화면용 글꼴이 서로 다른 기계에 존재하는 것이 가능하고, 따라서 X 클라이언트가 사용하는 모든 글꼴이 인쇄 가능하다는 보장이 없기 때문이다.

좋은 소식은 대부분의 WYSIWYG 응용 프로그램들이 이 문제에 대해 적당한 해법을 사용하고 있다는 것이다. 이 해법에는 화면 글꼴과 인쇄 글꼴 사이의 일종의 대응 관계를 만드는 것이 포함된다. (이 대응 관계가 주된 문제다. 이 밖에 굵은 변형꼴과 기울인 변형꼴, 로만 변형꼴을 하나의 글꼴 "족"으로 묶는 방법 등 다른 문제들도 있다. 불행히도, 이런 방법들에 표준이 없다. 이런 문제들을 해결하는 글꼴 관리 표준이 있다면, 모든 응용 프로그램들이 프로그램에 따라 다른 설정 대신 시스템 공통의 설정을 사용할 수 있으므로, WYSIWYG 인쇄 시스템에 글꼴을 설치하는 것은 훨씬 간단해 질 것이다.

7.2 애플릭스웨어(Applixware)

애플릭스웨어에 글꼴을 설치하는 데에는 두 가지 방법이 있다. 한 가지 방법은 애플릭스웨어 고유의 글꼴 서버인 폰타스틱을 사용하는 방법이다. 다른 하나는 이미 시스템에 설치되어 있는 글꼴을 사용하도록 애플릭스웨어의 fontmap을 편집하는 것이다. 글꼴 서버에 설치하는 쪽이 더 간편하지만, 이런 식으로 설치되는 글꼴은 300 dpi로만 인쇄될 수 있다.

7.2.1 폰타스틱(FontTastic)

폰타스틱을 사용하는 것이 글꼴을 설치하는 쉬운 방법이다. 새로운 글꼴을 설치하려면 단지 다음과 같이 하면 된다.

1. root로서 애플릭스웨어를 실행한다.
2. *tools* 메뉴 버튼을 클릭한다.
3. "Font Installer"를 선택한다.
4. 팝업 대화상자에서 "OK"에 표시한다.
5. "Catalogs" 메뉴를 클릭하고 "create"를 선택한다.
6. *catalog name* 박스를 채운다. 이 박스에 무엇을 써넣건 상관없다. 이 예에서는 "foobar"라고 했다고 가정하기로 한다.
7. 카탈로그 조작 목록에서 foobar 카탈로그를 선택한다.
8. "Services" 메뉴에서 "install fonts into -> FontTastic font server"를 선택한다.
9. 카탈로그 목록에서 foobar 카탈로그가 선택되어 있는지 확인한다. 다음에는 "select files" 버튼을 누른다.
10. 설치하려는 글꼴을 고르기 위해 select files 대화상자를 사용한다. 해당하는 파일을 선택한 뒤 "OK"를 누른다. 예컨데 /usr/share/fonts/ttfonts/ 디렉터리에서 arial.ttf를 선택하려면, "Current Directory" 대화상자 안에 */usr/share/fonts/ttfonts*를 입력하고 "OK"를 클릭한다. 여러 개의 파일을 선택할 수도 있지만, 모두 같은 디렉터리에 있어야 한다는 점에 주의하도록 한다.
11. 목록 박스 안의 글꼴을 체크하고, 필요하다면 박스 안의 글꼴을 삭제하거나 이름을 바꿈으로서 목록을 편집할 수 있다.
12. 준비가 끝났다면, "install fonts" 버튼을 클릭한다. 다음에는 "OK"를 클릭한다.
13. "services" 메뉴로 가서 "update"를 선택한다. annoying modal 대화상자에서 "OK"를 체크하고, services 메뉴에서 exit를 선택한다. 애플릭스웨어를 종료한다.
14. 작업 완료를 축하한다. 새로운 글꼴은 애플릭스를 다시 시작했을 때부터 사용가능할 것이다.

7.2.2 애플릭스웨어에서 시스템 글꼴을 사용하기

이 방법은 좀 더 복잡하지만, 더 좋은 결과를 얻을 수 있다. 글쓴이는 정말 중요한 글꼴, 많이 사용하는 글꼴이라면 이 방법을 사용하도록 권한다. 이 방법은 몇 가지 단계로 이루어진다.

X에서 글꼴을 사용할 수 있도록 한다. 이 단계에 대해서는 4 (앞 절)에서 설명했다.

고스트스크립트에서 글꼴을 사용할 수 있도록 한다. 이 단계에 대해서도 5.1 (앞 절)에서 설명했다.

fontmap.dir 편집한다. 이 단계가 애플릭스에서 글꼴을 사용할 수 있도록 하기 위한 마지막 단계이며, 아울러 가장 많은 시간이 필요한 단계이다. *fontmap.dir* 파일은 애플릭스가 설치된 디렉터리 아래의 *ax-data/fontmetrics*에 들어있다. 이 단계의 목적은 기본적으로 애플릭스에게 어느 스크린 글꼴이 어느 외곽선 파일에 대응되는지 알려주는 것이다. 스크린 글꼴이 항상 응용 프로그램이 설치된 컴퓨터에 있지는 않기 때문에, 이것은 그렇게 단순한 문제가 아니다.

여기서는 fontmap.dir에 글꼴을 추가하는 방법을 기술하기로 한다. 이 예에서는 Baskerville Italic 글꼴을 추가해 본다.

1. 먼저 FontRecord = Baskerville-Normal-Italic 라는 한 줄을 추가한다. 사실 FontRecord에 쓰는 글꼴 이름은 완전히 마음대로다. 하지만, 이 글꼴 이름은 한 글꼴에 하나씩 있어야 한다. 이 때문에, 고스트 스크립트에서 설치할 글꼴에 사용한 이름을 여기서도 사용하는 것이 좋은 습관이다.

2. 다음에는, Family = Baskerville 라는 한 줄을 적는다. 글꼴의 족(family) 이름은 애플릭스의 글꼴 선택 메뉴에 나타나는 이름이다. 대개 굵은 글꼴, 기울인 글꼴, 로만 글꼴, 굵은 기울인 글꼴 등의 변형꼴이 모두 원래의 글꼴과 같은 족에 속하기 때문에, *여러* 글꼴에 하나의 족 이름이 주어진다.

3. 글꼴이 굵은 글꼴이거나 기울인 글꼴, 굵은 기울임 글꼴이라면, 다음의 줄을 추가하여야 한다. 기울인 글꼴이면

   ```
   Slant = 1
   ```

   을 추가하고, 굵은 글꼴이면

   ```
   Weight = 1
   ```

   을 추가한다. 만일 굵은 기울인 글꼴이라면 두 줄 모두를 더한다. 이 예에서는 다음의 줄 만을 더한다.

   ```
   Slant = 1
   ```

4. 다음과 같은 줄을 더한다.

   ```
   ScreenName = "-paradise-baskerville-medium-i-normal--0-0-0-0-p-0-iso8859-1"
   ```

   ScreenName은 X 서버가 사용하는 글꼴 이름이다. "bask"라는 문자열을 포함하는 글꼴 이름들을 알아내려면

   ```
   xlsfonts|grep -i bask
   ```

   라고 입력하면 된다.

5. 이제 프린터 글꼴의 이름을 지시하는 줄을 다음과 같이 추가한다. PostScriptPrinterName = Baskerville-Normal-Italic

6. 다음에는 글꼴 메트릭 파일과 외곽선 파일의 위치를 지시해 주어야 한다.

   ```
   MetricsFile = /usr/share/fonts/misc/baskvli.afm
   Type1FontFileName = /usr/share/fonts/misc/baskvli.pfb
   ```

트루타입 파일을 추가한다면, 다음과 같이 afm 파일을 생성하기 위해 ttf2pt1을 사용할 수 있다.

```
ttf2pt1 -A foo.ttf - > foo.afm
```

(또는 ttfutils 꾸러미를 구해서 ttf2afm를 쓸 수도 있다.) 다음에는 다음과 같은 식으로 써준다.

```
MetricsFile = /usr/share/fonts/misc/foo.afm
```

Type1FontFileName 지시자를 포함해서는 *안된다*. 이 지시자는 고스트스크립트가 처리하도록 한다.

이상이다. 글꼴 일족을 모두 추가했다면, 다음과 같은 식으로 되어 있어야 한다.

```
FontRecord = Baskerville-Normal
Family = Baskerville
ScreenName = "-paradise-baskerville-medium-r-normal--0-0-0-0-p-0-iso8859-1"
PostScriptPrintName = Baskerville-Normal
MetricsFile = /usr/share/fonts/misc/baskvl.afm
Type1FontFileName = /usr/share/fonts/misc/baskvl.pfb

FontRecord = Baskerville-Normal-Italic
Family = Baskerville
Slant = 1
ScreenName = "-paradise-baskerville-medium-i-normal--0-0-0-0-p-0-iso8859-1"
PostScriptPrintName = Baskerville-Normal-Italic
MetricsFile = /usr/share/fonts/misc/baskvli.afm
Type1FontFileName = /usr/share/fonts/misc/baskvli.pfb

FontRecord = Baskerville-Bold
Family = Baskerville
Weight = 1
ScreenName = "-paradise-baskerville-bold-r-normal--0-0-0-0-p-0-iso8859-1"
PostScriptPrintName = Baskerville-Bold
MetricsFile = /usr/share/fonts/misc/baskvlb.afm
Type1FontFileName = /usr/share/fonts/misc/baskvlb.pfb

FontRecord = Baskerville-Bold-Italic
Family = Baskerville
Weight = 1
Slant = 1
ScreenName = "-paradise-baskerville-bold-i-normal--0-0-0-0-p-0-iso8859-1"
PostScriptPrintName = Baskerville-Bold-Italic
MetricsFile = /usr/share/fonts/misc/baskvlbi.afm
Type1FontFileName = /usr/share/fonts/misc/baskvlbi.pfb
```

이 설정 파일을 가지고 더 많은 일들을 할 수도 있다. 파일 자체에 설정 파일의 형식을 설명해주는 *용어 해설*이 들어있으므로 참고하라.

7.3 스타오피스(Star Office)

여기서는 스타오피스 5.0을 다룬다. 스타오피스 5.1에서도, 유틸리티가 psetup 대신 spadmin라고 불린다는 점을 제외하면, 절차는 비슷하다. 이 주제에 대해서는 *John McLaughlin's page* <http://www.mindspring.

com/~john_mcl/adding_fonts.html> 가 아주 훌륭한 자료원이며, 다음 내용의 대부분이 위 페이지에서 영감을 얻은 것이라는 점을 먼저 언급해 둔다.

스타오피스 5.0과 5.1 양쪽 모두 시도해 본 결과, 글쓴이는 스타오피스 5.1이 새 글꼴을 추가할 때 더 낫다는 느낌을 받았다. 스타오피스 5.0에는 트루타입 글꼴을 추가하는데 성공하지 못했지만, 스타오피스 5.1에서는 어느 정도 쉬웠다.

7.3.1 시작하기 전에 설정을 백업하자!

설정을 돌이킬 수 없도록 바꿀 경우에는 백업을 해 두는 것이 현명하다. 글꼴을 변경하는 것은 xp3 내의 여러 파일들에 영향을 준다. xp3/psstd.fonts 파일은 반드시 백업해야 하며, 더 나아가 xp3 디렉터리 전체를 백업하기를 권한다. 백업을 작성하려면 자신의 스타오피스 디렉터리로 가서

```
tar cvzf xp3.tgz xp3
```

를 실행하면 된다. 백업을 써서 원래 상태를 복구하려면, xp3 디렉터리를 삭제한 다음 압축 파일을 푼다.

```
rm -rf xp3
tar xvzf xp3.tgz
```

7.3.2 스타오피스에 타입1 글꼴 추가하기

타입1 글꼴을 스타오피스에 추가하는 것은 비교적 간단하다. 스타오피스 5.0에서 트루타입 글꼴을 사용하고 싶다면, 가장 좋은 방법은 타입1 글꼴로 변환한 뒤 여기에서 간단히 설명하는 과정을 따르는 것이다. 스타오피스 5.1을 갖고 있다면 (약간 더 어렵기는 하지만) 타입1으로 변환하는 대신 트루타입 글꼴을 설치하는 과정을 이용할 수도 있다. 먼저, 여느 때처럼 X와 고스트스크립트에서 글꼴을 쓸 수 있도록 한다. 이 작업이 끝나면, psetup을 사용하여 글꼴을 스타오피스에 설치할 수 있다. 과정은 다음과 같다.

1. root로서, psetup을 실행한다. (스타오피스 5.1을 가지고 있다면 spadmin을 실행한다.)
2. "add fonts" 버튼을 누른다.
3. 다음으로 넘어가는 가장 쉬운 방법은 "initialize font paths" 버튼을 누르는 것이다. 이렇게 하면 X 글꼴 패스 안에 있는 모든 글꼴 디렉터리의 목록이 스타오피스의 목록 박스에 나타난다.
4. 설치하려는 글꼴이 들어있는 디렉터리를 선택한 다음, "OK"를 누른다. (이 디렉터리는 박스 안에 있어야 한다.)
5. "convert all font metrics" 버튼을 클릭한다.

끝이다. 글꼴 설치를 완료했다. 스타오피스를 종료시켜도 된다. (아니면 종료할 때 까지 "OK"를 클릭하던지.) 스타오피스를 다시 시작하면, 새로운 글꼴을 사용할 수 있을 것이다.

7.3.3 스타오피스에 트루타입 글꼴 추가하기

스타오피스에 트루타입 글꼴을 추가하는 것은 간단한 일은 아니지만, 가능하다. 고역을 치르고, *John McLaughlin's page* <http://www.mindspring.com/~john_mcl/adding_fonts.html> 를 오랜 시간 읽은 끝에, 글쓴이는 마침내 스타오피스 5.1에서 트루타입 글꼴이 작동하도록 했다. 여기서 설명하는 것은 버전 5.0에서는 통하지 않는다는 것에 주의하기 바란다. 다음의 절차들은 고스트스크립트를 통해 인쇄를 하고 있을 경우, 적절하다.

- X에서 글꼴을 쓸 수 있도록 한다.
- 고스트스크립트에서 글꼴을 쓸 수 있도록 한다.
- 추가하려고 하는 글꼴의 afm 파일을 가지고 있어야 하므로, afm 파일을 만들기 위해

  ```
  ttf2pt1 -A foo.ttf - > foo.afm
  ```

  명령을 사용한다. 다른 방법으로는,
- *ttfutils* <http://www.moisty.org/~brion/linux/ttf.html> 꾸러미를 구해서 ttf2afm을 사용할 수도 있다. 이 방법의 장점은 여러 개의 글꼴을 다음의 예와 같이 한 번에 처리할 수 있다는 것이다.

  ```
  ttf2afm *.ttf
  ```

- 스타오피스에서는 각 ttf 파일에 대응하는 pfb 파일이 필요한데, 다음의 명령으로 만들 수 있다.

  ```
  touch foo.pfb
  ```

  사실 스타오피스는 이 파일을 인쇄를 위해서만 사용한다. 그리고 글꼴을 PPD 안에 집어넣어서 (고스트스크립트의 표현 체계 내에 실제로 글꼴이 들어있을 경우) 스타오피스가 글꼴이 프린터에 내장된 것으로 인식하도록 하고나면, 이 .pfb 파일을 사용할 필요가 없어진다. 스타오피스는 단지 글꼴을 설치하기 위해서 pfb 파일을 요구하는 것으로 보인다.
- 이제 글꼴을 설치하기 위해 spadmin을 실행시킬 수 있다.
- 이제 글꼴을 자신의 프린터 설정에 해당하는 PPD 파일에 추가한다. 여기서 적는 글꼴의 명칭은 고스트스크립트에서 사용한 명칭이 *아니라*, 스타오피스에서 사용한 명칭이어야 한다. 예를 들어, 글꼴이 foobar.ttf이고 대응하는 afm 파일은 foobar.afm이라면, PPD 파일에서 글꼴을 지시하기 위해서 "foobar"라는 이름을 써야 한다. 목록은 다음과 같은 꼴이어야 한다.

  ```
  *Font cloistrk: Standard "(001.002)" Standard ROM
  ```

한편, 고스트스크립트를 이용하여 인쇄하지 않는다면, 다룰 문제가 다르다. 이 경우, 프린터가 설치하려는 트루타입 글꼴을 가지고 있다고 스타오피스를 속이는 것은 좋은 생각이 아니다. 프린터의 ROM에는 그 글꼴이 없으므로, gv는 포스트스크립트 파일을 깨끗하게 보여주더라도 프린터는 그 파일을 프린트할 수가 없을 것이기 때문이다. 포스트스크립트 프린터를 가지고 있을 경우, 주된 차이점은 다음과 같다.

- PPD 파일을 편집하지 말 것.
- 크기 0인 pfb 파일을 만들기 위해 touch foo.pfb를 쓰면 안된다. 그 대신, 타입42 포스트스크립트 글꼴로 된 pfb 파일이 필요하다. 타입42 글꼴은 실은 "printer TrueType font"이다. 대부분의 응용 프로그램들이 타입42 글꼴을 알아서 처리하기 때문에, 이 형식의 글꼴을 쓰더라도 실제로 타입42 글꼴을 사용자가 인식하는 일은 거의 없다. 타입42 글꼴을 만들기 위해서는 *ttfps* <ftp://ftp.dcs.ed.ac.uk/pub/jek/programs/ttfps.tar.gz> 를 쓴다.

  ```
  ttfps foo.ttf foo.pfb
  ```

몇 가지 지저분한 것들(gotchas)이 있다. 스타오피스는 가끔 사용자가 원하는 스크린 글꼴을 사용하지 않을 때가 있다. 때때로 xp3/psstd.fonts를 확인할 필요가 있으며, 정말 스크린 표시용으로 마음에 두고 있는 글꼴을 스타오피스가 사용하는 것을 확실히 하기 위해 이 파일을 편집할 수도 있다. 사용하도록 또, 스타오피스는 설정 문제를 우아하게 다루지 못한다. 만약 설정에 잘못이 있으면, 워드 프로세서가 시작조차 못할 수도 있다. 때문에 xp3 디렉터리를 백업해 두어야 한다.

7.3.4 덮개 아래

트루타입 글꼴을 스타오피스에 설치하고 싶다면, 스타오피스가 작동하는 방식을 배울 필요가 있다. `spadmin`이나 `psetup`을 실행하면, 다음과 같은 일들이 일어난다.

- 스타오피스는 `xp3/pssoftfonts` 디렉터리안에 `pfb` 외곽선 파일로의 심볼릭 링크를 만든다.
- `afm` 파일이 `xp3/fontmetrics/afm/` 디렉터리 안으로 복사된다.
- 등록내용이 `xp3/psstd.fonts` 파일에 추가된다. 이 파일은 스타오피스가 사용하는 모든 스크린 글꼴의 이름을 저장하고 있다. (특히 스크린 글꼴을 외곽선 파일의 이름으로 대응시킨다.)

이것이 간단히 `xp3` 디렉터리 전체를 백업해 두는 것이 좋은 까닭이다. `xp3` 디렉터리의 백업은 스타오피스의 설정을 깨끗하게 복구하는 가장 손쉬운 방법이다.

7.4 워드퍼펙(Word Perfect)

아직 아무 것도 정리하지 못했다. 워드퍼펙에 글꼴을 설치하는 데 관해서는 *Rod Smith's webpage* `<http://www.rodsbooks.com/wpfonts/>` 가 훌륭한 자료원이다.

제 8 절 TeX / LaTeX

8.1 LaTeX/TeX 글꼴에 대한 간단한 입문

TeX과 LaTeX에 글꼴을 추가하는 것은 약간 복잡하다. 하지만 많은 일들이 그렇듯이, 어떻게 하는지 알고나면 이것도 쉬운 일이다. 어떤 글꼴들은 메타글꼴 형식으로, 어떤 글꼴들은 타입1 형식으로 배포된다. 보통 타입1 형식을 이용하는 편이 더 쉽다. 하지만 메타글꼴에는 크기에 따라 형태를 맞춘다는 뛰어난 장점이 있다. 이에 비해 타입1과 트루타입 글꼴의 경우, 크기가 다른 글꼴들은 똑같은 형태를 확대하거나 축소시킨 것일 뿐이다. 크기에 따라 글꼴의 형태가 달라지는 것이 좋은 주된 이유는, 크기 외의 조건들을 무시할 경우, 글꼴은 작은 크기에서는 폭이 비교적 넓어야 하고 크기가 클 때에는 폭이 좁아야 하기 때문이다.

이 절에서는 타입1 글꼴에 촛점을 맞추기로 한다. 타입1이 더 널리 쓰이고, 설치에 더 문제가 많기 때문이다.

LaTex 글꼴에 대해 간단히 알아보기로 하자. LaTeX은 타입1 글꼴을 처리하기 위해 아래와 같은 유형의 글꼴 파일들을 사용한다.

- `.pl` – 특성 목록(property list). 이것은 tex 글꼴 메트릭 파일을 사람이 읽을 수 있도록 한 것이다.
- `.vpl` – 가상 특성 목록(virtual property list). 가상 글꼴 파일을 사람이 읽을 수 있도록 한 것.
- `.fd` – 글꼴 정의(font definition). 글꼴의 *족(family)*을 정의하기 위해 사용된다.
- `.tfm` – TeX 글꼴 메트릭(tex font metric). 용어 해설에서 설명한 바와 같이 메트릭 파일이다. 이 파일은 타입1 글꼴이 사용하는 `.afm` 파일과 아주 닮았다(completely analogous).
- `.vf` – 가상 글꼴(virtual font). 이 파일에는 부호화에 대한 세부사항이 들어있으며, 해석기의 역할을 한다. TeX은 이 가상 글꼴들을 글꼴로서 다룬다. 예를 들어, 적은 수의 (이를테면 20개쯤) 본래 글자를 대신하는 글자(alternate character)로 이루어진 foobar-exp.pfb라는 희한한 글꼴이 있고, 이 대체 글자들 중의 일부를 사용하는 가상 글꼴이 있다고 해보자. (그리고 이 가상 글꼴은 나머지 글자들은 foobar.pfb라는 글꼴에서 가져온다고 하자.) dvips는 "난 foo.vf 가상 글꼴의 65번 글자가 필요해요"라고 할 것이다. dvips는 TeX 체계에서는 65번이 언제나 "a"라는 것을 알고 있다. 그러면 가상 글꼴은 TeX의 요구를 타입1 글꼴인 foobar.pfb의 14번 글자로 대응시켜 준다. (이 14번 글자는 타입1 글꼴

foobar.pfb에 있는 "a"의 대체 문자일 것이다.) 가상 글꼴 방식은 매우 유연해서 글꼴을 여러 다른 글꼴 파일들로부터 구성할 수 있도록 해준다. 이런 방식은 Adobe의 "전문가" 글꼴과 같은 글꼴을 사용할 때 편리하다.

- `.pk` – 장치에 따라 달라지는 비트맵 글꼴. 이 글꼴은 보통 필요해졌을 때 만들어진다. (이 글꼴은 타입1이나 메타글꼴을 해당 장치에서 표현한 것이다.) 이 글꼴은 대개 약 300-1200dpi 정도로 해상도가 높으며, 프린터에 출력할 목적으로 만들어진다. 높은 해상도와 크기에 따라 모두 별개의 `.pk` 파일이 필요하다는 점 때문에, 이 글꼴은 상당한 디스크 용량을 필요로 하게 된다. 때문에 잠시 저장되기는 하지만, 시간이 지나면 지워진다.
- `.mf` – 메타글꼴 파일. 메타글꼴은 글꼴 디자인에 널리 사용되는 그래픽 프로그래밍 언어이다. (그래픽에도 쓰일 수 있다.) 이 형식은 트루타입이나 타입1 방법에 비해 많은 장점들을 가지고 있다. 하지만 트루타입이나 타입1 만큼 흔하지 않다는 큰 약점이 있다. (그리고 WYSIWYG 출판에 아주 적합하지도 않다. 물론 TeX을 주된 조판 방식으로 사용한다면 이것은 큰 약점은 되지 않는다.)

TeX의 디렉터리 구조를 알아두면 좋다. 알아두어야 할 주요 디렉터리는 다음과 같다.

- `$TEXMF/fonts` – 주된 글꼴 디렉터리
- `$TEXMF/fonts/type1` – 타입1 글꼴 디렉터리
- `$TEXMF/fonts/type1/foundry` – 특정한 업체의 형태 파일 (shape file)이 들어가는 디렉터리
- `$TEXMF/fonts/type1/foundry/fontname` – *name*이라고 불리는 글꼴이 들어간다. *name*은 대개 평범한 영어 글꼴이며, TeX에서 쓰이는 복잡한 글꼴 명명 방식을 따를 필요가 없다.
- `$TEXMF/fonts/afm/foundry/fontname` – `foundry`라는 업체의 `name`이라는 글꼴에 해당하는 `afm` 파일이 들어있는 디렉터리.
- `$TEXMF/fonts/tfm/foundry/fontname` – `afm` 디렉터리와 같지만, 대신 `tfm` 파일이 들어있다.
- `$TEXMF/fonts/vf/foundry/fontname` – 위와 마찬가지이지만 가상 글꼴이 들어있다.
- `$TEXMF/fonts/source/foundry/fontname` – 위와 같지만, 메타글꼴 파일이 들어있다.
- `$TEXMF/dvips/config/psfonts.map` – dvips를 위한 fontmap 파일. 이 파일은 고스트스크립트의 Fontmap 파일과 기능과 형식 양면에서 비슷하다.
- `$TEXMF/tex/latex/psnfss` – 모든 글꼴 정의 파일들이 들어가는 디렉터리이다.

8.2 타입1 글꼴 추가하기

8.2.1 글꼴에 이름붙이기

먼저 글꼴에 적당한 이름을 붙여야 한다. 글꼴에 이름을 짓는 방법에 대한 안내는 자신의 시스템에 있는 fontinst 관련 문서를 보도록 하라. (이 문서는 tetex 문서가 들어있는 디렉터리의 fontinst라는 하위 디렉터리에 있을 것이다.) 아주 간략히 말하자면, 이름 붙이는 방법은 다음과 같다. `FNW{V}E{n}` :

- F는 업체를 한 글자로 나타내는 약자다. (m = monotype, p = adobe, b = bitstream, f = free)
- N은 글꼴 이름을 나타내는 두 글자의 약자이다. (예를 들어 ag는 "avant garde" 글꼴의 약자이다.)
- W는 글꼴의 두께이다. (r = regular, b = bold, l = light, d = demibold)
- V가 있을 경우 이는 기울인 변형꼴을 나타낸다. (i = italic , o = oblique)
- E는 부호화 방식의 약자이다. (대부분의 경우 Adobe의 표준 부호화 방식을 나타내는 `8a`이다.)

- n이 있을 경우 글꼴 폭 변형꼴을 나타낸다. (n은 폭을 좁힌 변형꼴)

예를 들어 Adobe의 Garamond demibold 글꼴의 이름은 pgad8a이다.

8.2.2 가상 글꼴과 tex 글꼴 메트릭 만들기

이제는 다음과 같이 fontinst를 수행할 수 있다.

```
latex `kpsewhich fontinst.sty`
```

그리고 나서 프롬프트에서 다음과 같이 입력한다.

```
\latinfamily{font_name}{}\bye
```

여기서 font_name은 글꼴 파일 이름의 맨앞 세 글자이다. (예를 들어 Adobe의 garamond 글꼴은 pad이다.) 이제 fontinst는 글꼴 기술(description) 파일, 특성 목록 파일, 가상 특성 목록 파일 등의 많은 파일들을 생성하게 된다. 많은 .mtx 파일들도 fontinst에 의해 만들어지지만, 이 파일들은 사용할 필요가 없다. 다음에는 특성 목록과 가상 특성 목록을 메트릭과 가상 글꼴로 변환시켜야 한다. 이 작업에는 vptovf와 pltotf라는 유틸리티를 사용한다.

```
for X in *.pl; do pltotf $X; done
for X in *.vpl; do vptovf $X; done
```

이제는 묵은 vpl, pl, mtx 파일들을 삭제한다.

8.2.3 dvips 설정하기

자신의 dvips의 설정 파일인 psfonts.map을 편집해야 할 것이다. 예를 드는 것이 이 파일의 형식을 설명하는 가장 좋은 방법이다.

```
marr8r      ArialMT <8r.enc <farr8a.pfa
marbi8r     Arial_BoldItalicMT <8r.enc <farbi8a.pfa
marb8r      Arial_BoldMT <8r.enc <farb8a.pfa
marri8r     Arial_ItalicMT <8r.enc <farri8a.pfa
marr8rn     Arial_Narrow <8r.enc <farr8an.pfa
```

8r.enc는 단지 dvips에게 사용된 부호화 방식을 알려주기 위해 있는 것이다. (fontinst가 가상 글꼴을 구성하는 방식 때문에, 이 문서의 모든 예에서는 부호화 방식이 8r이다.) 가장 왼쪽에 있는 열은 TeX이 사용하는 글꼴 이름이다. 두번째 열은 글꼴의 실제 이름으로서 글꼴 파일 안에 기록되어 있다. (이 이름은 afm 파일을 문서 편집기로 열어서 FontName 지시자를 찾아보면 알아낼 수 있다.) 마지막 열은 글꼴에 해당하는 형태 파일의 이름이다. 각 파일의 디렉터리는 알려줄 필요가 없다. tex은 파일을 찾아야 할 곳을 알고 있기 때문이다.

8.2.4 글꼴 시험하기

다음과 같은 문서를 latex으로 처리해보자.

```
\documentclass{article}
\begin{document}
        \usefont{T1}{pga}{m}{n}\selectfont
        \huge
        Testing a new font \dots the quick red fox jumped over the lazy brown dogs
\end{document}
```

여기에서 pga를 시험하려는 글꼴을 나타내는 약자로 바꾼다. latex이 제대로 처리하면, 글꼴 추가는 거의 끝난 것이다. 이제 할 일은 파일들을 정확한 디렉터리에 위치시키고(디렉터리는 입문 부분에서 설명했다), tex이 디렉터리 목록을 갱신하도록

```
texconfig rehash
```

를 수행하는 것 뿐이다.

8.2.5 .sty 파일 만들기

글꼴을 더 쉽게 사용할 수 있도록 .sty 파일을 만들고 싶을 수가 있다. 그렇다면, `$TEXMF/tex/latex/psnfss` 파일을 바탕양식으로 이용하도록 하라.

제 9 절 리눅스용 글꼴 구하기

9.1 트루타입

9.1.1 상업용 소프트웨어

트루타입 글꼴은 상업용 소프트웨어에 딸려오는 경우가 많고, 마이크로소프트 워드와 워드 퍼펙트 같은 꾸러미에는 많은 트루타입 글꼴이 포함되어 있다. 워드 퍼펙을 구하는 것이 많은 양의 글꼴을 얻는 쉬운 방법이다. (정말 돈이 없다면, 윈도우즈용의 워드 퍼펙의 옛날 버전을 사면 된다. CD에 들어있는 글꼴들을 읽을 수 있다.)

9.1.2 마이크로소프트의 글꼴을 다운받는 방법

마이크로소프트는 여러 가지 트루타입 글꼴을 무료로 쓸 수 있도록 했다. `.exe` 파일은 단순한 압축 파일이므로 `unzip`을 이용해서 풀어낼 수 있다. 다운로드는 *the download site* `<http://www.microsoft.com/truetype/fontpack/win.htm>` 에서 가능하다.

9.1.3 Luc의 웹페이지

Luc Devroye's webpage `<http://cgm.cs.mcgill.ca/~luc/originalfonts.html>` 에는 자유로이 구할 수 있는 글꼴이 있는 여러 사이트로의 링크가 있다. 이 글꼴들 중에는 "해적판 글꼴(warez fonts)"가 아니라 진짜 무료인 글꼴이 많이 있다는 점이 특징이다.

9.1.4 트루타입 글꼴이 있는 웹 사이트들

무료로 다운받을 수 있는 글꼴을 제공하는 웹 사이트들이 여럿 있다. 예를 들어, *the freeware connection* `<http://www.freewareconnection.com/fonts.html>`에는 많은 자료실로의 링크가 있다.

9.1.5 글꼴 전문업체

여러 글꼴 전문업체들이 트루타입 글꼴을 팔고 있다. 하지만 이 글꼴들은 상당히 비싸고, 같은 돈이면 더 나은 타입1 글꼴을 구할 수 있다. 여기 대해서는 타입1 글꼴을 다루는 절에서 더 이야기 하기로 하자. 트루타입 글꼴을 저렴한 가격에 팔고 있는 곳은 *buyfonts* <http://www.buyfonts.com>. 이다. 값싼 글꼴을 사기 전에 윤리에 대한 절을 읽어 보기 바란다.

9.2 타입1 글꼴과 메타글꼴

9.2.1 맥과 윈도즈 형식을 다루기

많은 글꼴 전문업체들이 윈도즈와 맥 사용자들을 염두에 두고 글꼴을 시판하고 있다. 때로는 이것이 문제가 된다. 보통 "윈도즈 글꼴"은 `zip` 파일로 묶여있기 때문에 상당히 다루기 쉽다. 단지 `pfm` 파일을 `afm` 파일로 변환시키기만 하면 된다. (pfm2afm을 쓰면 된다.)

매킨토시 글꼴은 좀더 어렵다. 이 파일들은 `.sit.bin` 라는 확장자를 갖는 stuffit 압축 형식으로 만들어져 있는 경우가 보통이기 때문이다. 불행히도 리눅스에는 새로운 버젼의 stuffit로 만들어진 stuffit 압축을 풀 수 있는 도구가 없다. 유일한 방법은 맥 에뮬레이터인 Executor를 사용하거나 dosemu, Wine 등에서 stuffit를 실행해 보는 것이다. 일단 `sit.bin` 파일을 풀고나면, `t1utils` 꾸러미에 들어있는 을 사용하여 맥 파일들을 변환시킬 수 있다.

불행히도 몇몇 업체들은 타입1 글꼴을 매킨토시 형식(stuffit 압축)으로만 배포하고 있다. 하지만 글꼴 전문가인 *Luc Devroye* <http://cgm.cs.mcgill.ca/~luc/>에 의하면, 대형 글꼴 전문업체들은 모두 타입1 글꼴을 맥과 윈도즈 양쪽 다 사용할 수 있도록 하고 있다.

9.2.2 무료 글꼴

ctan <http://www.ctan.org>에는 좋은 글꼴들이 많이 있으며, 그 가운데 많은 것들이 무료다. 이 글꼴들 가운데 일부는 타입1 글꼴이지만, 대부분이 메타글꼴 형식으로 되어 있다. 타입1 버젼의 computer modern 글꼴을 만든 *Bluesky* <http://www.bluesky.com> 도 살펴보면 좋다. (computer modern 글꼴은 최상의 품질을 갖고 있다. 이 글꼴과 비슷한 품질과 완결성을 가진 다른 글꼴을 구입하려면 적어도 500 달러 정도가 들 것이다. computer modern은 최고급 글꼴들과 비견할 만 하다.)

Luc Devroye의 웹페이지 <http://cgm.cs.mcgill.ca/~luc/originalfonts.html> 에도 다운 받을 수 있는 무료 글꼴들이 있는 여러 사이트로의 링크가 있다. 이 글꼴들의 특징은 이 중에는 "해적판 글꼴"이 아니라, 진짜 공짜인 글꼴들이 많다는 점이다.

URW는 대부분의 프린터에 들어 있는 표준 포스트스크립트 글꼴을 공개 글꼴로 발표했다. 이 글꼴은 상당히 훌륭하다.

Walnut Creek Archive <ftp://ftp.cdrom.com/pub/os2/fonts/> 에 무료로 이용할 수 있는 글꼴들과 쉐어웨어 글꼴들이 여럿 있다. 이 중 몇 가지는 틀림없는 해적판이고, 별로 좋지 않은 것들도 있다. 어떤 식으로든 라이선스가 글꼴에 따라오지 않았다면, 해적판일 가능성이 있다. *Winsite* <http://www.winsite.com/win3/fonts/atm/> 의 windows 3.x 소프트웨어에 있는 fonts/atm 디렉터리에도 타입1 글꼴들이 여럿 있다. 안타깝게도, 이 글꼴들 중의 몇 개는 잘못된 afm 파일들을 가지고 있으며 커닝 짝들을 모두 잃어버린 상태다. (afm 파일의 "FontName" 절을 편집해서 afm 파일들을 수정할 수 있다. 수정 내용은 글꼴 형태 파일에서 주어진 글꼴 이름과 일치해야 한다. 물론 커닝 짝을 추가하는 것은 이 문서의 범위를 넘어서는 내용이다.)

Luc Devroye's webpage <http://cgm.cs.mcgill.ca/~luc/> 에는 많은 링크 뿐 아니라 그 자신이 디자인한 무료 글꼴들도 여럿 있으며, 인쇄에 관한 멋진 논의도 실려있다. 이 사이트는 "반드시 들러야 할 곳"이다. 이 사이트에는 많은 글꼴 업체들로의 링크도 있다.

9.2.3 상용 글꼴(Commercial Fonts)

일반 대 고급(Value vs Premium) : 왜 고급 글꼴을 사야 할까? 왜 어떤 글꼴은 비싸고 다른 글꼴은 쌀까, 궁금하지 않은가? 비싼 글꼴은 대부분의 포스트스크립트 프린터에 내장되는 "표준 포스트스크립트 글꼴"이고, 유명한 것들이다. 더 비싼 것들을 사야할 필요가 있을까? 글쓴이의 의견은 일반 사용자에게는 Bitstream CD에 들어있는 것들 같은 정도의 글꼴로 충분하다는 것이다. 하지만 글꼴을 "진짜 업무"에 쓴다거나 진짜 글꼴 매니아라면 좀더 좋은 글꼴은 필수품이다. 그리고 대부분의 고품질 글꼴은 Computer Modern 처럼 공개된 것이거나, 고급품 시장 지향의 상용 글꼴이다.

저렴한 글꼴의 장점은 자명하다. 싸다는 점. 하지만 고품질 글꼴도 나름의 장점이 있다.

- *윤리적인 문제:* 싼 글꼴은 해적판일 경우가 매우 많다. 서체를 디자인하는 데에는 긴 시간과 숙련된 디자이너가 필요하다. 하나 당 1달러 이하로 팔리는 글꼴은 파는 업체에서 디자인한 것이 아닐 확률이 100%에 가깝다. 비정상적인 양의 글꼴이 들어있는 CD는 해적판임이 거의 확실하다. (대규모 업체가 파는 수천 달러짜리 글꼴 모음집은 예외다.) 대개 해적판은 훌륭한 업체의 것에 비해 글꼴의 질이 떨어진다.
- *완벽성:* 고품질의 글꼴은 (특히 Adobe의 제품은) 사용자에게 보다 완벽한 글꼴 족을 제공하기 위해 멋진 보충 글꼴 등, 여러 가지 변형꼴을 포함하고 있다. 흔히 글꼴에는 굵은 글꼴, 기울인 글꼴, 약간 굵은 글꼴, 붓글씨체 대문자, 작은 대문자, 예스러운 숫자(old style figures), 추가 모은자 등의 보충 글꼴이 있다. 최근, Adobe는 사용자에게 하나의 글꼴 족 안에서 (거의) 무한정의 변형꼴을 제공하는 multiple master technology를 가지고 있다.
- *글꼴의 질:* 무료로 사용할 수 있는 글꼴 중의 많은 수와 싸구려 해적판은 커닝 짝과 적당한 모은자 같이 정말로 필요한 부분을 갖추고 있지 않다. 이런 글꼴은 기본적으로 싸구려다. 이에 비해, 훌륭한 디자이너들은 원래의 디자인을 연구하기 위해 많은 수고를 하며, 최선을 다해 그것을 새로운 것으로 바꾸어낸다.
- *진실성:* Adobe Garamond를 디자인한 사람(Robert Slimbash)은 실제로 원래의 서체를 디자인한 Claude Garamond를 연구했다. 사실 이름높은 글꼴 업체들은 휙휙 그물질해서 뭔가를 건진 다음 글꼴 그리기 전문가Fontographer)를 동원해 고치기 보다는, 언제나 여러 디자인들을 주의 깊게 연구하고 있는 것이다.

일반(Value)

- 괜찮은 품질의 여러 가지 타입1 글꼴들이 들어있는 CD를 구하기에 아주 좋은 곳이 *Bitstream* `<http://www.bitstream.com>` 이다. 유명한 Bitstreams의 제품으로는 *250 font CD* `<http://www.bitstream.com/products/world/font_cd/bits_collection.html>` 와 *500 font CD* `<http://www.bitstream.com/products/world/font_cd/500_cd.html>` 가 있다. (500 글꼴 CD는 이 글을 쓰는 시점에서 50 달러 이상이다.) 이 CD들에는 상당히 좋은 품질의 글꼴이 들어있어서, 일반 사용자들에게는 아주 좋은 출발점이다. Corel의 제품에 쓰이고 있는 글꼴들이 (대부분) Bitstream에게 라이선스를 얻은 것이다.
- *Matchfonts* `<http://www.matchfonts.com/>`는 좀더 저렴한 가격의 글꼴들을 제공하고 있다. 이 글꼴은 30달러에 약 8가지가 든 "꾸러미" 형태로 배포된다. 이 꾸러미에는 좋은 붓글씨체 글꼴들이 몇 가지 들어있다. 이 글꼴들은 모두 일반적인 형태, 즉 .exe 파일 하나로 된 윈도즈 ATM 글꼴로 제공된다. 하지만 확장자에 속아서는 안된다. 이건 그저 zip 압축파일일 뿐이다. 이 글꼴들은 글쓴이가 아는 한, 해적판이 아니다.
- *EFF* `<http://www.buyfonts.com>`는 건 당 최하 2달러 부터 트루타입 글꼴을 판다. 이 회사는 서체당 16달러 이상인 "전문가 영역"의 포스트스크립트와 트루타입 글꼴들도 가지고 있다.

고급(Premium)

- Adobe는 여러 가지 고품질의 글꼴을 가지고 있으며, *Adobe's type website* `<http://www.adobe.com/type/>` 에서 글꼴을 구입할 수 있다. 몇 가지는 비싸지만, 비교적 알맞은 가격의 번들도 몇 가지 있다. *Adobe Type Collections* `<http://www.adobe.com/type/collections.html>` 를 살펴 보라. Adobe는 시장에서 가장 완벽한 글꼴족을 몇 가지 가지고 있는데, 예를 들면 *Garamond* `<http://www.adobe.com/type/browser/P/P_912.html>`, *Caslon* `<http://www.adobe.com/type/browser/P/P_180.html>`, 과 *multiple masters* `<http://www.adobe.com/type/browser/C/C_4e.htm>` 가 있다. (Adobe의 웹사이트에서 쓰이고 있는 Myriad와 Minion이 Adobe의 multiple master 글꼴 가운데 훌륭한 것들이다.)
- *Berthold Types Limited* `<http://www.bertholdtypes.com>` 는 대형 글꼴업체로 여러 가지 고품질 글꼴을 제공하고 있다. 이 가운데 몇 가지는 Adobe를 통해 팔고 있지만, 모두 Berthold로부터 직접 구입할 수 있다. Adobe와 같이 만족할 만한 가격(price ballpark)이다.
- ITC는 Corel의 제품에 들어있는 것들을 포함해서 몇 가지 고품질 글꼴을 개발했으며, *http://www.itcfonts.com* `<http://www.itcfonts.com>` 에서 살펴볼 수 있다. ITC는 약 100-180 미국 달러에 글꼴족 꾸러미를 제공한다. ITC 글꼴은 타입1과 트루타입 형식이 함께 제공된다. "Windows" 꾸러미를 선택하는 편이 더 좋다. 맥 형식은 리눅스에서 처리하기 곤란하기 때문이다.
- *Linotype* `<http://www.linotypelibrary.com>` 는 Herman Zapf를 포함한 전설적인 디자이너들이 디자인한 글꼴들을 제공하기로 잘 알려진 업체이다. (이 사람은 나중에 "Zapf Chancery"라고 불리게 되었다. Palatino도 그가 디자인했다.)
- *Monotype* `<http://www.monotype.com>` 는 Microsoft 제품에 들어있는 글꼴의 대부분을 개발했다. 오래되고 잘 알려진 글꼴 업체 중의 하나이다.
- *Tiro Typeworks* `<http://www.portal.ca/~tiro/>` 는 조금 비싼 서체라면 좋은 품질의 것들을 팔고 있다. 이 업체의 서체는 매우 완벽하다. 이를테면, 완벽한 모은자 집합과 작은 대문자, 제목 글꼴 등이 들어있다. 유닉스는 OS 옵션 중의 하나로 열거되어 있다. "윈도즈 혹은 맥"이라는 낱말을 한참 본 다음에 이 옵션을 보니, 반갑기까지 하다.

그밖의 링크(More Links) 다른 업체들로의 링크가 필요하면 *Luc Devroye's page* `<http://cgm.cs.mcgill.ca/~luc/>` 로 찾아가보라.

제 10 절 쓸모있는 리눅스용 글꼴 소프트웨어

리눅스용 글꼴 꾸러미들이 여럿 있다. 이 가운데에는 반드시 있어야 할 것들이 많다.

- chkfontpath는 `xfs` 설정 파일을 관리하기 위한 유틸리티다.
- *DTM – the Definitive Type Manager* `<http://www.debian.org/~fog/dtm/>`는 시스템 전체의 글꼴을 관리하는 도구다. 이 프로그램은 개발자 판이다.
- *fontinst* `<http://www.tug.org/applications/fontinst/index.html>`는 LaTeX에 타입1 글꼴을 설치하는 것을 간단하게 해 주는 LaTeX 꾸러미이다.
- *Freetype* `<http://www.freetype.org>` 는 대부분의 리눅스 배포판에 들어있는 트루타입 라이브러리이다.
- *Ghostscript* `<http://www.cs.wisc.edu/~ghost/>` 는 리눅스에서 인쇄를 하기 위해 쓰이는 소프트웨어이다. 리눅스에 들어있는 고스트스크립트는 GNU 버전의 고스트스크립트이다. GNU 버전은 알라딘 고스트스크립트의 최신판 보다 한 버전 묵은 것이다. (알라딘 쪽은 묵은 버전의 고스트스크립트를 GPL로 공개하고 있다.)

- *pfm2afm* <http://pegasus.rutgers.edu/~elflord/font_howto/pfm2afm.tgz> 는 윈도즈용 pfm 글꼴 메트릭 파일을 리눅스에서 사용할 수 있는 afm 메트릭으로 변환하는 유틸리티이다. 이 버젼은 CTAN에서 구할 수 있는 원래의 버젼을 바탕으로, 리눅스에서 컴파일할 수 있도록 Rod Smith가 바꾼 부분을 포함하고 있다.
- *mminstance and t1utils* <http://www.lcdf.org/~eddietwo/type/>는 타입1 글꼴을 다루기 위한 두 가지 꾸러미이다. mminstance는 Adobe의 *multiple master* <http://www.adobe.com/type/browser/C/C_4e.html> 타입1 글꼴을 다루기 위한 것이고, t1utils는 서로 다른 타입1 형식들 사이의 변환을 위한 유틸리티 모음이다.
- *ttf2pt1* <http://quadrant.netspace.net.au/ttf2pt1/>는 트루타입을 타입1 글꼴로 변환한다. 타입1 글꼴이 필요한 응용 프로그램을 가지고 있다면 쓸모있다.
- *ttfps* <ftp://ftp.dcs.ed.ac.uk/pub/jek/programs/ttfps.tar.gz>는 .ttf 트루타입 글꼴 파일을 타입42 파일로 변환한다.
- *ttfutils* <http://www.moisty.org/~brion/linux/ttf.html> 트루타입 글꼴을 다루기 위한 유틸리티 꾸러미. 이 꾸러미는 ttf2pt1이 필요하다. 꼭 필요하지 않더라도 쓸모있는 꾸러미.
- *type1inst* <ftp://ftp.metalab.unc.edu/pub/Linux/X11/xutils/> 는 타입1 글꼴을 설치하기 위해서 필수적인 꾸러미이다. 이 꾸러미는 설치를 엄청나게 간단하게 해 준다.
- *xfstt* <ftp://ftp.metalab.unc.edu/pub/Linux/X11/fonts/> 는 리눅스용 트루타입 글꼴 서버이다. 이 프로그램도 쓸모있지만, xfs가 아마 더 나은 선택일 것이다.
- *xfsft* <http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/> xfsft 글꼴 서버. 이 프로그램은 xfs 안에 포함되어 있다는 것에 주의하자.
- *x-tt* <http://hawk.ise.chuo-u.ac.jp/student/person/tshiozak/x-tt/> 는 한국어와 일본어 글꼴을 처리하기 위해 고안된 글꼴 서버이다.

제 11 절 글꼴에 관련된 윤리 및 라이선스 문제

글꼴의 라이선스는 매우 말썽 많은 문제이다. 무료로 *이용할 수 있는* 글꼴들이 많이 있음에도 불구하고, 만일 그 글꼴에 별도로 지정하는 라이선스가 없다면, 그 글꼴은 어떤 의미에서 "도둑질"일 가능성이 있다. 이 문제는 서체에 대한 지적 재산권 법에 의해 한층 더 혼란스러워진다. 기본적으로 미국에서는 글꼴 *파일*은 저작권에 의해 보호되지만, *글꼴 표현(font renderings)*은 보호되지 않는다. 즉, 글꼴을 재배포(redistribute)하는 것은 불법이지만, 그 글꼴들을 그래픽용지에 인쇄해서 "역처리"하고 그 인쇄물에 맞춰 글꼴을 디자인하는 것은 완전히 합법적이다. 역처리된 글꼴은 보통 싼 값이거나 무료로 쓸 수 있지만, 질이 떨어진다. 이런 글꼴들은 흔히 해적판 글꼴들과 함께 엄청난 양의 글꼴이 들어있는 싼 CD로 배포된다. 따라서 어떤 글꼴이 역처리된 것인지, 그저 해적판인지 말하기란 늘 쉽지 않다. 이런 상황은 리눅스용으로 무료 글꼴 꾸러미를 만들고 싶어하는 사람들에게 엄청난 골칫거리가 되고 있다.

아마도 글꼴 도둑질의 본성 가운데 가장 야비한 것 중의 하나가 그 도둑질이 글꼴 디자이너가 한 작업의 가치를 떨어트린다는 점일 것이다. 해적행위를 당한 글꼴은 응당히 원래의 디자이너에게 돌려져야 할 감사도 없이, 늘 많은 수의 글꼴이 들어있는 CD 한 장에 번들로 끼워넣어지곤 한다. 이에 비해, 합법적인 글꼴 업체들이 칭찬받을 만한 점은 이 업체들은 글꼴 디자이너에 대한 감사를 명기하고 있다는 것이다.

이 문제에 대해서는 여러 가지 다른 의견들이 있다. 지적 재산권에 대한 설명은 *typeright* <http://www.typeright.org> for 을 보도록 하라. 다른 의견으로는 *Southern Software, Inc* <http://www.ssifonts.com/> 을 보라. 하지만 그들의 글꼴은 아무 것도 사지 말라! 이 회사의 타입1 글꼴은 Adobe 글꼴을 조잡하게 역처리한 것으로, AFM 파일이 없으며, 따라서 쓸모없다.

The comp.fonts FAQ <http://www.faqs.org/faqs/fonts-faq/part2/> 와 *Luc Devroye's homepage* <http://cgm.cs.mcgill.ca/~luc/> 에서도 글꼴과 지적 재산권의 문제를 토론하고 있다. 이 참고 자료들의 관점은 앞서의 사이트들에 비해 덜 극단적이다.

제 12 절 참고자료

12.1 글꼴 정보

- *Rod Smith's homepage* <http://www.rodsbooks.com/> 에는 애플릭스웨어와 워드 퍼펙에서 글꼴과 프린터를 사용하는 데 대한 정보가 듬뿍 담겨있다.
- *John McLaughlin's page* <http://www.mindspring.com/~john_mcl/adding_fonts.html> 는 스타오피스에서 글꼴을 설치하는 데 대해 이야기하고 있다.
- *Jim Land's homepage* <http://www.geocities.com/SiliconValley/5682/postscript.html> 는 포스트스크립트와 글꼴에 대한 사이트로의 링크가 많이 있다.
- *The comp.fonts FAQ* <http://www.faqs.org/faqs/fonts-faq/> 는 확실한 글꼴 FAQ이다.
- *Luc Devroye's homepage* <http://cgm.cs.mcgill.ca/~luc/> 는 글꼴 등등에 대한 정보가 배가 가라앉을 만큼 잔뜩 있다. 이 친구는 공개 글꼴을 한 웅큼 디자인했고, 이 홈페이지에는 재미있는 링크와 정보, 조언이 많이 있다.
- *The Font Deuglification HOWTO* <http://www.linuxdoc.org/HOWTO/mini/FDU.html> 는 리눅스에서 트루타입 글꼴을 쓰는 데 대해 논의하고 있다. 이 문서는 여러 "TrueType" 하우투 가운데에서도 발군으로, 아주 훌륭한 정보원이다.
- *TrueType Fonts in Debian mini-HOWTO* <http://www.dimensional.com/~bgiles/debian-tt.html> 는 데비안에서 트루타입을 설치하는 데 대해 논의하고 있다. 데비안 사용자라면 반드시 읽어보아야 한다. 트루타입을 지원하는 버젼의 xfs가 들어있는 배포판을 쓰고 있지 않다면, 역시 읽어볼 가치가 있다.
- *The (preliminary) True Type HOWTO* <http://www.moisty.org/~brion/linux/TrueType-HOWTO.html> – 1998년 6월 자의 불완전한 하우투. 이 목록에는 가능한 정보를 모두 싣기 위한 목적으로 포함시켰다.
- *TrueType for XFree86 Mini-HOWTO* <http://www.sfu.ca/~yzhang/linux/truetype/> – 약간 오래된 하우투. 레드햇 5.x에만 적용될 수 있다.

12.2 포스트스크립트와 인쇄 정보

- *Adobe's Postscript page* <http://www.adobe.com/print/postscript/main.html> 는 포스트스크립트 표준에 대한 확실한 사이트다.
- *Ghostscript's home page* <http://www.cs.wisc.edu/~ghost/> 에는 많은 정보와 모든 최신 프린터 드라이버가 있다.
- *Jim Land's homepage* <http://www.geocities.com/SiliconValley/5682/postscript.html> 에는 포스트스크립트와 글꼴에 대한 사이트로의 링크가 많이 있다.
- *Christopher Browne's Printing FAQ* <http://www.hex.net/~cbbrowne/printing.html>

제 13 절 용어 해설

- **afm Adobe Font Metric**을 나타낸다. 이 파일에는 글꼴의 형태에 대한 정보와는 반대로, 글꼴의 폭과 간격에 대한 정보가 들어 있다.

- **anti-aliasing** 은 font smoothing이라고도 하는데, 모니터와 같이 해상도가 낮은 장치에서 글꼴을 표시하기 위해 사용되는 기술이다. 글꼴을 표시하는데 있어서, 문제는 글꼴은 외곽선으로 이루어지는 반면, 장치는 점으로 표시한다는 것에 있다. 글꼴을 표시하는 단순한 방법은 외곽선의 안쪽에 있는 픽셀들은 검게 칠하고, 나머지 점들은 건드리지 않고 두는 것이다. 이런 방식에는 외곽선 위에 놓인 픽셀을 적절하게 처리할 수 없다는 문제가 있다. 보다 현명한 알고리듬이라면 경계에 있는 픽셀을 회색으로 칠해야 하는데, Anti-aliasing에는 이런 알고리듬이 필수적으로 포함되어 있다.

- **bdf** 글꼴은 X에서 사용할 수 있는 여러 가지 비트맵 글꼴이다.

- **비트맵 글꼴(bitmap fonts)** 이 글꼴은 단순한 점들의 집합이다. 글꼴의 각 글자는 점들의 행렬(matrix)로 저장된다. 따라서 비트맵 글꼴은 장치에 따라 달라지고, 스크린과 프린터에서 같은 비트맵 글꼴을 쓸 수 없게 된다. 스크린 비트맵 글꼴의 예로는 X에서 사용되는 .pcf와 .bdf 글꼴이 있고, 프린터 비트맵 글꼴의 예에는 TeX의 PK 글꼴이 있다.

- **didone** 모던(modern)을 보시오.

- **DPI** 인치 당 도트수. 모니터는 보통 75-100 DPI로 표현하고, 신형 프린터는 300에서 1200 DPI 사이이다.

- **전문가 글꼴(expert fonts)** 은 보통의 글꼴을 보조하는 추가 글꼴 모음이다. 여기에는 작은 대문자 글꼴, 장식 글꼴, 기본 모은자 이외의 모은자, 가변폭 숫자 등이 포함된다. Adobe의 글꼴 중에는 사용할 수 있는 전문가 글꼴을 가진 것들이 많이 있다.

- **글꼴 서버(font server)** XFree86에서 글꼴을 사용할 수 있도록 하는 백그라운드 프로그램.

- **글리프(glyph)** '형태(shape)'를 멋부려서 쓰는 말이 glyph이다. 이것은 외곽선 글꼴을 구성하는 한 요소이다. 예를 들어 "i"라는 글자의 점, 수직선, 세리프 등이 모두 glyph다. glyph는 글꼴의 형태를 결정한다.

- **커닝(kerning)** 가변폭 글꼴의 경우, 이웃하는 글자 짝에 따라 글자 사이의 간격이 다르다. 글꼴 메트릭 파일에는 **커닝 짝(kerning pairs)**이라고 하는, 글자 짝 사이의 간격에 대한 정보가 저장되어 있다.

- **모은자(ligature)** 모은자란 일련의 문자들을 나타내기 위해 사용되는 특별한 문자이다. 모은자는 예를 들어 설명하는 것이 가장 이해하기 쉽다. fi 라는 글자를 표시하는 경우, "i"의 윗점은 "f"와 충돌하고, "i"의 왼쪽 위에 달리는 세리프도 "f"의 수평획과 충돌할 수 있다. fi의 모은자는 f자 뒤에 i자가 따라올 때 쓸 수 있는 하나로 된 활자이다. fl, ffi, ffl의 모은자도 있다. 대부분의 글꼴에는 fi와 fl의 모은자만 들어있다. **전문가 글꼴**에서는 다른 모은자들도 사용할 수 있다. 이어진 글자들을 예로 들어 설명하자.

- **메타글꼴(metafont)** 글꼴을 만들기 위해 사용되는 그래픽 언어. 메타글꼴은 많은 장점을 가지고 있는데, 그 가운데 중요한 것이 메타글꼴에 의해 만들어진 글꼴은 크기 변화가 반드시 똑같은 비율로 이루어질 필요가 없다는 점이다. 예컨데, 메타글꼴로부터 생성된 17 포인트의 computer modern 글꼴은 10 포인트의 computer modern 글꼴을 확대한 것과 같지 않다. Adobe의 multiple master technology에 앞서 이런 면을 갖는다는 것이 메타글꼴의 특징이다. 메타글꼴의 주된 장점은 고품질의 글꼴을 만들어 낸다는 것이다. 약점은 외곽선 글꼴로부터 비트맵을 만들어 내는 것이 느리다는 것으로, 이 때문에 메타글꼴은 WYSIWYG 출판에는 적합하지 않다.

- **메트릭(metric)** 여기에는 하나의 글꼴이 얼마나 공간을 차지할지에 대한 정보가 저장되어 있다. 하나의 글꼴 메트릭은 그안에 글꼴을 채워넣을 수 있는 상자 같은 것이다. 문서(page)에 글자들을 늘어놓기 위해서는, 글꼴의 형태 자체는 필요하지 않지만, 글꼴 메트릭은 반드시 필요하다. 때문에 보통 가변폭 글꼴은 형태에 대한 정보와 더불어 메트릭의 정보도 가지고 있다. 메트릭에는 커닝에 대한 정보도 포함된다.

- **모던(modern)** 글꼴은 19세기 이후 개발된 디자인에 바탕한 글꼴이다. 모던은 수직선을 강조하기 때문에 강건한 외양을 갖는다. 모던은 올드스타일이나 트래디셔널에 비해 더 "정직(character)"하고 "단정(attitude)"하지만, 어느 정도의 위엄과 형식성도 가지고 있다. 모던은 긴 글을 쓰기에는 적절하지 않지만, 짧은 글을 적기에는 쓸모있다. Bodoni가 유명한 모던 서체이다.
- **올드스타일(old style)** 글꼴은 전통적인 서체의 글꼴이다. 올드스타일 글꼴은 15세기 후반까지 거슬러 올라가는 디자인에 바탕하고 있다. 올드스타일 글꼴은 책처럼 긴 문서를 쓰기에 매우 훌륭하다. 올드스타일 글꼴은 초기 디자이너들의 전통 속에서 디자인된 것이지만, 상당히 근래에 디자인된 올드스타일 글꼴도 있다. 특히 **Goudy Old Style**체는 20세기 초에 Goudy가 디자인한 것이다. 중요한 올드스타일 체로는 Goudy Old Style, Garamond, Caslon 등이 있다.
- **pcf** 글꼴은 X가 사용하는 비트맵 글꼴이다.
- **포스트스크립트(postscript)** 는 문서(page)를 기술하기 위해 고안된 프로그래밍 언어이다. 포스트스크립트는 그것을 고안한 Adove의 상표였지만, ISO의 표준이기도 하다. 포스트스크립트를 표현하기 위해서는 해석기가 필요하다. 이 작업은 고스트스크립트 처럼 컴퓨터에 있는 프로그램에 의해 이루어지거나, 일부 포스트스크립트를 지원하는 프린터에 의해서 처리된다.
- **세리프(serif)** 글꼴은 세리프라고 불리우는 작은 갈고리가 글꼴의 끝 부분에 붙은 글꼴이다. 세리프는 보통 글꼴을 보다 읽기 좋게 해준다. 하지만 세리프는 해상도가 낮은 장치에서는 매우 표현하기 어렵고, 특히 글꼴의 크기가 작을 경우에는 가늘고 작은 세리프는 잘 나타나지 않는다. 때문에 해상도가 낮은 장치에서 작은 크기의 글꼴을 쓸 경우에는 마이크로소프트의 Verdana 같은 산세리프 글꼴이 더 읽기 좋은 경우가 많다. 또 한가지 지적해 둘 점은 긴 문서를 쓰기 위해 디자인되지 않은 modern 같은 산세리프 글꼴들이 있다는 점이다.
- **산세리프(sans serif)** 글꼴은 세리프가 없는 글꼴이다. (sans는 "없는"이라는 뜻의 프랑스어다.) 이 글꼴은 빳빳한 모양을 하고 있으며, 중요한 제목을 쓰는데 적합하다. 교과서 인쇄에서는 산세리프 글꼴을 중요한 제목에만 쓰도록 하고 있지만, 다른 곳에도 산세리프를 쓸 수 있다. 강한 인상 보다는 읽기 좋은 쪽에 무게를 두고 디자인된 산세리프 글꼴들도 있다. 카탈로그나 판매용 팸플릿 처럼 간단히 훑어보게 되는 짧고 박력이 필요한 문서에 산세리프를 쓸 수 있으며, 최근에는 마이크로소프트가 해상도가 낮은 장치에서 작은 크기일 때 읽기 쉽도록 디자인된 Verdana 글꼴을 내놓았다. 잘 알려진 산세리프 글꼴에는 Lucida Sans, MS Comic Sans, Avant Garde, Arial, Verdana, Century Gothic이 있다.
- **슬래브세리프(slab serif)** 는 세리프가 마치 벽돌(slab) 처럼 보이고 갈고리 꼴이 없는 글꼴의 한 갈래이다. 슬래브세리프 글꼴은 언제나 그런 것은 아니지만 아주 읽기 좋은 경우가 많다. 세리프가 단순하고 강하기 때문에, 이 글꼴은 읽는 사람에게 종이에 구멍이 뚫려있는 듯한 느낌을 준다. 잘 알려진 슬래브세리프의 예를 들자면, Clarendon과 New Century Schoolbook, Memphis 등이 있다.
- **트래디셔널(transitional)**글꼴은 올드스타일 글꼴 보다 나중의 디자인에 바탕한 글꼴이다. 트래디셔널 글꼴 가운데에는 읽기 좋은 것들이 많이 있다. 중요한 트래디셔널 글꼴로는 Baskerville과 Times Roman이 있다.
- **타입1(Type1)** 는 Adobe에 의해 디자인된 글꼴 종류이다. 이 글꼴은 X 서버의 아키텍쳐와 포스트스크립트 표준에 의해 오래 전부터 지원되었기 때문에, 거의 모든 리눅스 응용 프로그램들이 타입1을 잘 지원한다. 포스트스크립트 글꼴은 여러 가지 형식으로 배포된다. 유닉스용 포스트스크립트 글꼴은 흔히 `afm` (adobe font metric) 파일 하나와 외곽선 파일 하나로 배포된다. 외곽선 파일은 보통 `.pfb`(printer font binary)이나 `.pfa` (printer font ascii) 파일이다. 글리프는 모두 외곽선 파일에 들어있고, 메트릭 파일은 메트릭을 포함하고 있다.
- **타입3(type3)** 글꼴은 타입1과 비슷하다. 파일의 확장자도 `pfa`와 `afm`로 타입1 글꼴과 비슷하지만, X에서는 지원되지 않는다. 이 때문에 이 글꼴을 지원하는 리눅스 응용 프로그램은 매우 드물다.