유닉스상의 한글 출력

최준호, junker@jazz.snu.ac.kr

이 강의에서는, 유닉스(리눅스 포함) 환경에서의 한글 입출력, 특히 출력에 대한 문제를 다룹니다.

목 록

1 유닉스상의 한글처리

유닉스에서 한글을 처리할 수 있는 방법에는 여러가지 방법이 있습니다. 크게는 국제화(I18N, internationalization)이라는 방법과 지역화(L10N, localization)이라는 방법이 존재합니다. 국제화는 해당 소프트웨어(OS를 포함)을 여러 국가와 언어, 문화의 차이에 대응할 수 있도록 바꾸는 일이고, 지역화라는 것은 특정 국가, 지역, 언어, 문화 관습에 맞도록 프로그램을 바꾸는 일을 의미합니다. 보통 이런 문제는 크게 언어와 연관이 됩니다만, 한 국가에서 여러 언어를 사용하는 경우도 있고, 여러 나라에서 한 언어를 사용하는 경우도 있으며, 하나의 언어라도 지역적인 특성에 따라 차이가 날 수도 있으므로 반드시 그러한 것은 아닙니다.

유닉스 OS의 국제화를 실현하기 위해 여러 표준을 제정하는 기구들이 노력을 하였습니다. 이런 표준 중 하나가 로케일(locale)이라 부르는 것입니다. 유닉스 시스템의 로케일은, 사용자와 명령의 수행 환경을 제어할 수 있는 환경변수와 C라이브러리의 집합을 제공하여 프로그램의 별다른 수정 없이 여러 언어와 지역, 그리고 문화의 차이를 보완할 수 있습니다.

한글의 경우, 지역의 문제는 그리 크지 않습니다. 하지만 이를 처리하는 데에는 상당한 노력이 필요합니다. 이는 크게

- 한글의 전자적인 표현

- 한글의 입력
- 한글의 출력
- 기존 프로그램과의 조화

등의 문제로 나누어지게 됩니다. 이 문제와 관련하여, 프로그램상의 다국어 메시지 지원도 문제가 될 수 있습니다.

1.1 한글의 전자적인 표현

한글의 전자적인 표현은, 한글을 컴퓨터 안에서 어떤 방식으로 저장하는가에 따른 문제입니다. 보통 한글은 2바이트 코드로 나타내는 것이 일반적인데, 유닉스 상에서 가장 많이 사용하는 방식은 완성형(KSC5601)코드입니다. 완성형 코드는 한글의 모든 조합 가능한 글자들을 표현할 수 없다는 단점이 있지만, 유닉스에서 사용하는 EUC(Extended Unix Charset)코드에 잘 부합합니다. 조합형의 경우 2바이트 중 두번째 코드가 일반적인 제어문자와 보통 문자의 영역과 충돌하는 관계로 유닉스에서 사용하기는 적합하지 않습니다. 보통 많이 쓰이는 인코딩 방식으로는 다음과 같은 것이 있습니다.

- EUC-KR(KSC5601완성형, 2바이트, 0xa1-0xfe, 0xa1-0xfe)
- ISO-2022-KR(KSC5601완성형, 2바이트, 0x21-0x7e, 0x21-0x7e)

이 두가지 코드는 모두 완성형 코드를 나타낼 수 있다는 점에서는 동일하지만, 비트의 표현 방식이 다릅니다. EUC-KR은 우리가 보통 생각하는 MSB가 1로 세트된 완성형 코드이고, ISO-2022-KR은 일반적으로 한글 전자우편 전송에 사용하는 방식으로, MSB를 0으로 사용하지만 이 경우 기존의 영문자 영역(US-ASCII)와 겹치는 관계로 한글을 나타내는 경우에는 앞뒤로 구분할 수 있는 제어문자를 붙여주게 됩니다. 최근에는 ISO에서 제정한 유니코드에 한글의 조합 가능한 모든 글자가 포함되어 있고, Windows NT나 Java등에서 유니코드를 지원하고 있으므로, 앞으로는 한글의 모든 글자를 표현하기 위한 인코딩 방법으로 조합형보다는 유니코드가 쓰이게 될 것이라 생각합니다.

1.2 한글의 입력

한글 입력은 어느 환경에서 한글을 보아야 하는지에 따라 달라집니다. 텍스트 기반의 전용 터미널을 사용하는 경우 출력을 위해서는 터미널 자체가 한글을 입출력할 수 있는 기능이 있어야 하며, 그래픽을 사용할 수 있는 경우에는 별도로 그려 주어야 하는 기능이 있어야 합니다. 한글 입력은 보통 자판의 문제로 생각되는데, 2벌식과 3벌식이 있습니다. 이는 국제화의 경우에도 별도로 정해진 규정은 없으며, 대부분의 경우 필요한 소프트웨어에 한글을 입력할 수 있도록 입력 부분을 가로채는 방식으로 이루어집니다. 이런 방법을 사용한 것이 PC 콘솔에서는 han이라는 프로그램이며, X윈도우에서는 HanX와 X용 한글 입력기로 나타납니다.

han은 원래 일본어를 PC콘솔에서 나타나게 하기 위한 프로그램인 kon(Kanji ON console)에 한글 입력기능을 덧붙인 것입니다. kon은 기본적으로 일본어 뿐 아니라 한글 출력 기능도 가지고 있었습니다.

HanX는 X윈도우의 기반이 되는 X11라이브러리 자체를 조작하여 입출력 부분에 한글 입출력 루틴을 집어넣은 것입니다. 이를 공유 라이브러리 방식으로 활용하면 기존의 프로그램을 바이너리 상태로 그대로 한글 입출력이 되도록 바꿀 수 있습니다만(MS윈도우의 한글화 방식과 비슷한 점이 있습니다), X윈도우의 국제화를 위해 제공하는 국제화된 입출력 방식과는 관계가 없기 때문에 표준을 무시한다는 점이 있으며, 국제화된 입력 방식과 충돌의 우려가 있다는 것이 단점입니다.

X용 한글 입력기의 필요성은 오래전부터 제기되어 왔습니다. X11R5에서 제대로 된 형태로 나타난 국제화 방식은, X11R6에서 표준으로 정착되어 X11라이브러리의 한 축을 이루게 되었습니다. X11R5에서는 국제화에 대한 규정만 있었을 뿐, 구현 방법은 각자에게 맡겨두었기 때문에 크게 Xsi와 Ximp라는 두가지 방법이 존재하였습니다. 이 두가지 방법은 국제화된 입력기 - 영어권이 아닌 언어로 X11에서 입력을 하고자 하는 경우 - 에 대한 두가지 프로토콜을 제공을 하였습니다. 이 문제는 X11R6에서 하나의 국제화된 표준을 지키는 것으로 해결

되었고, Ximp에 바탕을 둔(그러나 다른) XIM프로토콜이 만들어졌습니다. XIM은 X윈도우에서 다국어 입력을 위한 X서버와의 프로토콜을 지키는 클라이언트(X입력기)를 만들기 위한 기반을 제공합니다. 약 1년 전부터 '벼루'와 'KIMS', 'hanIM'등이 만들어져 한글 입력에 도움을 주고 있습니다. X에서 한글을 입력해야 하는 가장 빈번한 경우는 Netscape의 입력창입니다. 따라서 입력기의 가장 큰 문제는 Netscape에서 자유로운 한글 입력이 되어야 한다는 것인데, 아직 이 문제를 만족스럽게 해결한 입력기는 없는 것으로 보입니다.

별도로, hanterm이나 hanemacs와 같은 프로그램은 직접 프로그램 자체를 수정하여(이 경우에는 xterm와 GNU emacs) 한글 입출력 기능을 내장한 예입니다. 이는 아마도 X11에서 비교적 자유로운 글꼴의 입출력이 가능하였기 때문이라고 생각합니다. 이들은 모든 한글을 표현하기 위하여 별도로 이야기 글꼴에 바탕을 둔 조합형 글꼴을 바탕으로 두고 있습니다.

2 한글의 출력

한글 출력의 경우 가장 큰 문제는 정해진 방법이나 지켜야 할 표준이 없다는 것입니다. 따라서 한글 출력을 보조해 주는 각자의 프로그램은 나름대로의 방법으로 한글 출력을 지원합니다만, 공통적인 것은 모두 결과적으로 포스트스크립트 프로그램을 출력한다는 것입니다.

2.1 콘솔에서의 출력

콘솔과 같은 텍스트만의 환경인 경우에는 그 터미널이 한글 출력을 지원할 수 있는지가 큰 문제가 됩니다. Linux/FreeBSD의 경우 han을 사용하여 콘솔에서 한글 출력을 볼 수 있습니다만 han은 안정되어 있는 소프트웨어는 아닙니다. 이외에 리눅스용 콘솔 드라이버를 한글화 하려는 노력이 있으나, 현재 큰 진전은 없는 상태입니다.

2.2 X윈도우에서의 출력

출력의 대상을 조금 더 넓히면 X윈도우의 출력을 다루어야 합니다. X윈도우는 Type1글꼴을 사용하여 윤곽선 글꼴의 출력을 지원하지만, 단지 화면에 나타나도록 하는데에만 매달렸을 뿐, 이를 프린터로 출력하는 문제에 대해서는 간과하고 말았습니다.

X윈도우는 R4부터 2바이트 문자를 갖는 언어를 위한 출력 함수를 지원해 왔고, 이를 R5에서는 FontSet이라는 개념으로, R6에서는 이를 더욱 확장하여 XOM(X Output Method)라는 개념으로 지원을 하고 있습니다. 로케일과 연계되어, 이들은 각 언어를 출력할 수 있는 방법을 제공하여, 한글의 경우 R5이후부터의 국제화 방식에 잘 대응하고 있습니다. X11R5에서는 보통 완성형 글꼴(다만 ISO-2022-KR처럼 MSB가 0인)을 많이 사용하는데(XLFD에서 -ksc5601.1987-0으로 끝납니다), 이를 사용하면 X11에서의 출력은 국제화 규정을 지키며 프로그래밍하는 경우에 기존 프로그램이라도 간단한 리소스 조작만으로 원하는 효과를 얻을 수 있습니다.

이를 잘 지키는 예로는 Netscape가 대표적입니다. 이는 국제화 규정을 잘 활용하여 X11환경에서 다양한 언어를 표현할 수 있도록 하였습니다. 이와 반대로, hanterm이나 hanemacs등은 한글 출력의 문제를 프로그램에서 내장한 형태입니다. 이들은 주로 조합 글꼴을 사용하는데, 조합 글꼴은 X11에서 보통 사용하는 한글 완성형 글꼴과는 달리 자소를 조합하여 한글을 출력하므로 크기도 작고 모든 글자를 나타낼 수 있지만 표준적인 방법에서 처리하는 것이 어렵다는 단점이 있습니다.

2.3 출력물

그러나 이는 단지 화면 상에서만의 문제이고, 이를 프린터로 출력하고자 하는 때에는 또 다른 방법이 필요합니다. X11은 국제화된 화면 출력은 지원하지만, 프린터를 통한 출력에는 특별히 정해진 표준이 없습니다. XPrint라는 방식이 있는데, 이는 널리 쓰이는 것이 아닙니다. MS윈도우즈의 경우 트루타입이라는 윤곽선 글꼴을 사용하여 화면용의 글꼴과 출력용의 글꼴을 일치하여 사용할 수 있지만, X윈도우의 경우 기본적인 글꼴이 모두 비트맵 형식으로 사용되기 때문에, 그리고 윤곽선을 지원하지만 한글을 사용할 수 있는 윤곽선 폰트를

지원하지 않는 관계로, 출력에서 있어서는 모두 별도의 방식을 사용해 왔습니다. 이들은 여러가지 프로그램이 존재합니다만, 다음과 같은 프로그램이 대표적입니다.

- 조판: hTeXp/hLaTeXp, HLaTeX
- 텍스트 출력: hpscat, h2ps, nh2ps
- Netscape: nhpf, nhppf
- Ghostscript: gs-hfftype

양질의 문서를 얻을 수 있는 조판(typesetting) 프로그램인 TeX은, 주로 이를 많이 사용하는 이공계 계열 사람들에 의해 한글화가 진행되어 왔습니다. KAIST수학과의 hTeXp/hLaTeXp는 가장 처음으로 만들어진 한글 TeX과 그 역사를 같이 하고 있습니다. 이 프로그램은 TeX의 에러 메시지에서도 한글 문서를 확인할 수 있는 등의 장점을 가지고 있지만, 사용하는 글꼴의 저작권 문제로 출력에 필요한 DPI(300, 600등)의 비트맵 형태(pk글꼴)로만 제공된다는 것이 유일한 단점입니다. 하지만 일반적인 환경에서 사용하는데에는 무리가 없습니다. HLaTeX은 TeX의 매크로 패키지인 LaTeX에서 한글을 구현하도록 한 LaTeX의 보조 패키지로 사용할 수 있는데, 저자의 많은 노력의 결과로 공개된 형태의 여러가지 한글 글꼴을 제공합니다. 이 글꼴의 일부는 문화부에서 공개한 바탕체 등의 글꼴에 기반을 두고 있습니다.

일반적인 경우, 유닉스에서는 대부분 양질의 출력물을 얻기 위한 별도의 시스템을 갖고 있지 못합니다. 이를 위해서는 포스트스크립트 언어를 지원하는 프린터로 포스트스크립트 파일을 출력하는 경우가 많은데, 포스트스크립트 언어를 사용하는 프린터는 보통 고가이고, 별도의 옵션을 장착해야 하는 경우가 많으므로, ghostscript라는 포스트스크립트 해석기를 사용하여 프린터 뿐 아니라 화면 출력이나 이미지 등으로 포스트스크립트 프로그램의 출력 결과를 저장할 수 있습니다.

유닉스에서 많이 사용하는 일반 텍스트 출력을 작성하기 위해서는 먼저 어떤 글꼴을 사용할 것인가의 문제가 해결되어야 합니다. 각 프로그램을 이를 위해서 나름대로의 해법을 제공합니다. hpscat의 경우에는 ked에서 사용하던 조합 방식의 포스트스크립트 글꼴과 2바이트 한글 포스트스크립트 글꼴을, h2ps에서는 n바이트 방식의 포스트스크립트 글꼴을, nh2ps의 경우 HLaTeX에서 사용하는 완성형 방식의 한글 포스트스크립트 글꼴과 2바이트의 한글 포스트스크립트 글꼴을 모두 사용할 수 있습니다.

포스트스크립트 글꼴은 윤곽선 글꼴이며, 힌팅을 통해 낮은 해상도의 비트맵을 변환하여도 글자가 제 모양대로 보일 수 있다는 장점을 가지고 있지만, 영어 표현을 위해 널리 사용하는 Type 1 글꼴의 경우 한 글꼴에 256자만을 사용할 수 있다는 단점이 있습니다. 실제로 HLaTeX의 경우, 한글 포스트스크립트 글꼴은 256자의 한 글꼴 여러 벌을 사용하여 한글 글꼴 한 벌을 만들고 있습니다. 완성형의 경우 2350자이므로, 한글의 표현에는 10벌의 Type1 글꼴이 사용되며, 한자와 전각 기호를 감안하면 이 수는 더욱 많아집니다. nh2ps에서는 이 글꼴을 사용하기 위해 완성형 한글 코드를 HLaTeX의 한글 글꼴에 맞는 글꼴 번호와 그 인덱스로 변환해 주는 표를 이용하고 있습니다. 이 방법을 해결하기 위한 것이 Type 0 “Composite” 글꼴(OCF, Original Composite Font)입니다. 이는 여러개의 Type 1이나 Type 3 글꼴을 모아 계층적인 형태로 2바이트의 글꼴을 사용할 수 있도록 해줍니다. 현재 상용으로 구현되어 있는 2바이트의 한글 글꼴은 이 방식을 취하고 있습니다. 이 글꼴의 문제는 글꼴의 크기가 크다는 것인데, 이는 대부분의 포스트스크립트 지원 프린터에서 미리 다운로드 할 수 없는 크기이며, 공개된 Type 0의 2바이트(8/8) 형태의 한글 글꼴이 존재하지 않습니다. 포스트스크립트 언어를 만들어 냈던 Adobe사는 CID-Keyed Font라는 새로운 글꼴 형식을 제안했는데, 이는 인코딩에 상관없이 글꼴을 제작할 수 있도록 하여 하나의 글꼴이 여러 인코딩에 사용할 수 있도록 하는 개념을 바탕으로 하고 있습니다. 바탕체를 이용한 공개 버전도 나와 있으며, 하나의 글꼴을 이용하여 유니코드, 완성형, 조합형 등의 여러가지 인코딩으로 사용할 수 있습니다.

hpscat은 독특한 한글 조합형 포스트스크립트 글꼴을 사용합니다. 이는 hpscat의 출력으로 나오는 포스트스크립트 프로그램 이전에 프린터에 로딩되어야 할 필요가 있는데, ghostscript를 사용하는 환경에서는 단지 먼저 글꼴을 출력하는 것만으로 충분합니다. 이 방법을 위해서는 미리 ghostscript를 수정할 수도 있고, h2ps와 같이 매번 글꼴을 출력하게 할 수도 있습니다. hpscat은 별도로 2바이트 한글 포스트스크립트 글꼴을 사용할 수 있습니다.

h2ps는 독특한 n바이트 한글 코드(ncode)에 기반을 둔 한글 포스트스크립트 글꼴을 사용합니다. 이는 크기가 작고 가독성이 높으며, 출력을 위해 별도의 조합 과정이나 여러 벌의 자소가 필요하지 않다는 장점이 있지만, 고정폭 글꼴을 제공하지 못하고, 널리 쓰이지 않는 코드를 사용한다는 장점이 있지만, 별도의 글꼴 로딩 없이 어느 포스트스크립트 프린터에서나 사용할 수 있는 프로그램이기도 합니다.

nh2ps는 HLaTeX의 미려한 한글 글꼴을 이용하고자 만들어진 프로그램이며, 기존의 a2ps에 기반을 두고 있습니다. 현재에는 Type 0의 한글 글꼴도 지원하도록 하고 있습니다. nh2ps를 사용하여 한글 문서를 출력하려면, 미리 ghostscript에서 글꼴을 읽을 수 있도록 등록을 해 주거나, 글꼴을 미리 다운로드 해 주어야 합니다. 글꼴을 다운로드 하는 작업은 쉽지 않은 관계로, 대부분의 경우 ghostscript를 통해 처리해 주어야 합니다.

유닉스에서 인터넷을 위해 가장 많이 사용하는 프로그램일 Netscape에서는 한글 문서의 출력을 지원합니다. 인코딩을 한글로 맞추고 출력을 하면 되지만, 여기에는 몇가지 문제가 있습니다.

- Save as...의 포스트스크립트 형식으로 저장하는 경우, 글꼴의 모양(이탤릭이나 볼드)등을 구별하지만, 한글 글꼴과 영문 글꼴을 나누지 않고 하나로 취급합니다. 따라서 이 경우 별도의 필터를 거치지 않으면 한글을 볼 수는 없습니다.
- Print...의 경우 기본적으로 포스트스크립트 형식의 출력을 하는데, 한글 의 경우 Type 0의 한글 글꼴을 사용하여 출력을 해 줍니다. 그러나 한글의 경우 글꼴의 모양은 모두 한가지인 것으로 생각해 버립니다. 따라서 한글의 경우에 기울어진 글꼴이나 굵은 글꼴을 볼 수가 없습니다. 하지만 해당 한글 글꼴을 처리할 수 있는 포스트스크립트 편집기가 있는 경우 한글 문서를 큰 문제없이 출력할 수 있습니다.

이 문제를 해결하기 위해 nhpf와 nhppf라는 두가지의 프로그램이 존재합니다. 이 두 프로그램 모두 Save as...와 Print...의 포스트스크립트 출력 결과를 한글을 출력할 수 있는 형태로 잘 바꾸어 줍니다만, nhpf는 h2ps에서 사용하는 n바이트 한글 글꼴을 사용하고, KSC5601에서 한글 부분만 지원하며, nhppf는 nh2ps와 비슷한 제약조건을 갖지만 KSC5601의 모든 글자를 나타낼 수 있습니다.

하지만, Print로 저장한 포스트스크립트 결과물의 경우 반드시 nhpf/nhppf를 거쳐야 할 필요는 없습니다. Type 0의 한글 글꼴을 사용할 수 있다면 문제가 없습니다만, 앞에서 말한 이유로 공개된 버전이 없다면 대안은 CID-Keyed Font밖에는 없지만, 이것도 마찬가지로 공개 버전에서는 한글만을 이용할 수 있다는 단점이 있습니다.

다른 방향으로 생각해 보면, 양질의 다른 글꼴을 포스트스크립트 글꼴로 사용할 수 있다면 문제는 달라질 것입니다. 그래서 만들어진 것이 gs-hfftype입니다. 이는 ghostscript를 수정하여 MS의 트루타입 한글 글꼴을 ghostscript를 사용하여 Type 0 한글 글꼴로 에뮬레이션을 하는 gs의 패치입니다. 이는 두가지의 큰 성과에 도움을 얻고 있는데, 하나는 공개 트루타입 래스터라이저인 FreeType Project이고, 다른 하나는 이를 사용하여 ghostscript에서 일본어를 처리하고자 한 gs-kfftype입니다. hfftype은 기본적으로 이 gs-kfftype에 기반을 두고 있으며, 한글 인코딩과 한글 글꼴 처리를 위한 부분을 추가한 것입니다. 이를 사용하면 트루타입 글꼴을 이용하여 Netscape의 문서를 출력할 수도 있고, Type 0 글꼴을 지원하는 hpscat이나 nh2ps에서 보다 다양한 글꼴을 사용하는 것이 가능합니다.

위의 논의와는 별도로, Adobe의 전자문서 포맷인 PDF(Portable Document Format)이 있습니다. 한글이 섞인 문서를 만들어 내는 것은 HLaTeX등을 사용하여 ghostscript를 사용하면 가능하지만, pdf의 특징 중 하나인 효율성을 살리지 못하는 경우가 많습니다. 한글 pdf를 만들려는 노력이 있습니다만, 아직 MS윈도우 환경에서 그치고 있습니다.

2.4 글꼴

X윈도우에서나 출력물을 위해서나, Linux등의 공개 유닉스에서 꼭 필요한 것이 공개된 한글 글꼴입니다. 잘 알려진 공개 X11용 글꼴에는 한양 글꼴과 조금 제한되기는 하지만 한양시스템의 웹바탕 등이 있습니다. X11R5이후는 기본적인 한글 글꼴로 대우가 기증한 명조와 고딕의 두가지가 있지만, 크기와 모양이 다양하지 못하다는 것이 단점입니다. 이 문제를 해결하기 위해서는

- 공개된 글꼴의 증가

- 다른 글꼴 포맷의 변형

의 두가지 방법이 있습니다. HLaTeX의 한글 글꼴을 X의 비트맵으로 변환하는 SERI 글꼴의 예는 공개된 글꼴을 증가하고자 하는 노력이지만 손이 많이 가는 작업입니다. 다르게는 트루타입과 같은 다른 윤곽선 글꼴을 X11용 글꼴로 변환하는 방법도 생각해 볼 수 있습니다. ttf2bdf-k가 대표적인 예인데, FreeType Project를 이용하여 만들어진 프로그램 중에 ttf2bdf라는 트루타입을 X에서 사용하는 비트맵 글꼴로 바꾸어 주는 프로그램을 수정하여 한글 트루타입 글꼴에서 bdf포맷을 추출하는 일이 가능합니다.

출력물의 경우도 그리 다르지는 않아서, 공개된 CID-Keyed Font와 HLaTeX의 글꼴을 제외하면 양질의 글꼴이 많은 편은 아닙니다. 다만 현재는 gs-hfftype의 도움으로 공개된 한글 트루타입 글꼴(대부분 한글만 갖고 있지만)을 사용할 수 있습니다.

트루타입 글꼴을 사용할 때 가장 주의해야 하는 것이 바로 저작권 문제입니다. 이 글꼴을 다른 OS, 특히 유닉스에서 사용할 수 있는지, 정해진 용도가 아닌 다른 용도로 사용해도 되는지에 관한 문제입니다. 대략적으로 볼 때 Bitstream이 Cyberbit가 가장 적당한 글꼴이라고 생각합니다.

3 끝으로

대부분의 한글 어플리케이션은 hanterm와 같은 몇몇 예를 제외하고는 점점 국제화 추세에 따르는 방향으로 나아가고 있습니다. 하지만 여전히 출력의 문제에 대해서는 국제화된 노력이 미미하고, 따라서 각자 나름대로의 방법으로 문제를 해결하려 하고 있기 때문에 발전이 더딘 점이 많습니다. 포스트스크립트는 잘 정리되어 있고, 출력의 목적으로 사용하기에는 더없이 좋은 언어이지만, 글꼴의 문제로 적극적으로 활용되지 못하는 단점이 있습니다. 앞으로는 양질의 공개 글꼴이 더 많이 늘어나고, X11과 출력 문서를 위해 공통적인 글꼴을 사용할 수 있도록 윤곽선 형태의 글꼴의 증가와, 글꼴 서버등을 통한 X11용 글꼴의 선택이 증가되었으면 하는 바램입니다.

4 참고 자료

4.1 인터넷

Internationalization Concept Dictionary

<http://cns-web.bu.edu/pub/djohnson/web_files/i18n/i18n.html##index>

Home Page of Ulrich Drepper

<http://i44s11.info.uni-karlsruhe.de:80/~drepper/>

LANG=ko 이야기

<http://www3.shinbiro.com/~teodeul/LANGko.html>

Unicode and Hangul

<http://camis.kaist.ac.kr/~jwjung/seminar/hangul-i18n>

Code and Hangul [한글과 코드]

<http://suny.multi.co.kr/~leesl/code/>

Unicode Consortium

<http://www.unicode.org>

CAIR 한글 프로그램 archive(han, HanX, hanterm, hanemacs)

<ftp://cair-archive.kaist.ac.kr/pub/hangul>

KIMS

<http://brain.tgmi.com/kims.html>

hanIM

<http://mizi.co.kr>

벼루

<http://entropy.kaist.ac.kr/~jhpark/byeoroo/>

한글 TeX 인터넷 홈페이지

<http://knot.kaist.ac.kr/htex/>

HLaTeX(CTAN mirror)

<ftp://ftp.kreonet.re.kr/pub/CTAN/language/korean/HLaTeX>

Ghostscript

<http://www.cs.wisc.edu/~ghost>

Adobe Corporation

<http://www.adobe.com>

SERI 한글 글꼴

<ftp://sensor.kaist.ac.kr/pub/fonts/>

한글과 Printing

<http://nuclina.hoseo.ac.kr/ps/>

김범철님의 "X Font"

<http://brain.tgmi.co.kr/xfont.html>

Ghostscript와 한글

<http://jazz.snu.ac.kr/~junker/gs-ko/>

CID-Keyed Font

<http://www.adobe.com/supportservice/devrelations/typeforum/cidfonts.html>

Hangul and Internet FAQ 21: How can I print out hangul document from Unix host?

<http://pantheon.cis.yale.edu/~jshin/faq/qa21.html>

h2ps

<http://cglab.snu.ac.kr/~yjlee/n3f/applications/h2ps.html>

nh2ps

<http://jazz.snu.ac.kr/~junker/work/nh2ps/>

Netscape Navigator

<http://home.netscape.com>

nhpf

<http://cglab.snu.ac.kr/~yjlee/n3f/applications/nhpf.html>

nhppf

<http://jazz.snu.ac.kr/~junker/work/nhppf/>

FreeType을 이용한 Ghostscript에서의 한글처리

`<http://jazz.snu.ac.kr/~junker/work/gs-ko/gs-ko-freetype.html>`

Adobe PDF

`<http://www.adobe.com/prodindex/acrobat/adobepdf.html>`

4.2 출판물

1. Adobe Corporation, PostScript Language Reference Manual, 2nd edition
2. O'Reilly & Associates, Programmer's Supplement for Release 6