

# TeX, LaTeX 소개

김강수\*

2012년 11월 26일

## 요약

이 글은 2007년 초에 작성하였던 것이다. 어느 학술지에 발표했던 글의 바탕이 된 것으로 TeX을 소개하기 위한 목적으로 썼던 글이었다. 당시는 아직 ko.TeX이 발표되기 전이었던 시점으로서 지금과는 텍 이용 환경에 많은 차이가 있었으므로 낡은 내용이 되어 버린 것이 많다. 내용의 대의를 유지하면서 그런 낡은 내용을 걷어내는 정도로 수정한다.

## 차례

1	Introduction . . . . .	2
2	TeX이란? . . . . .	2
2.1	엔진 . . . . .	3
2.2	매크로 . . . . .	4
2.3	폰트 . . . . .	4
2.4	유틸리티 . . . . .	6
3	TeX 사용의 몇 가지 측면 . . . . .	6
3.1	기본적인 LaTeX 문서 원본의 구조 . . . . .	6
3.2	TeX으로 문서작성하는 것의 효용 . . . . .	8
3.3	TeX의 다른 활용 가능성 . . . . .	8
4	KTUG과 TeX에서의 한글 사용 . . . . .	9
5	결어 . . . . .	10

---

\*필명: Nova De Hi

## 1 Introduction

위키백과사전<sup>1</sup>에는 TeX을 다음과 같이 소개하고 있다.

TeX (그리스어 발음으로  $\tau\epsilon\chi$ (테흐), 영어에서는 흔히 /tek/(텍)으로 읽는다. 그림 1처럼 출력하지 못하는 경우 TeX과 같이 가운데 ‘E’를 소문자로 적는다)은 Donald Knuth가 만든 조판 시스템이다. 폰트 기술언어인 METAFONT 및 컴퓨터 모던 글꼴과 함께 텍은 두 가지 목적을 염두에 두고 개발되었다. 첫째, 누구든지 적당한 노력을 들이면 고품질의 서적을 생산할 수 있게 하려는 것이다. 둘째, 어떤 컴퓨터에서든, 현재와 미래에도 완전히 동일한 결과물을 생성하는 시스템을 만들려는 것이다.

이 프로그램은 자유(free)이고, 주로 학술활동과 관련하여 특히 수학, 물리학, 컴퓨터과학, 경제학, 정치학 및 공학에서 널리 쓰인다.

TeX이 복잡한 수학적식을 조판하는 가장 좋은 방법이라고 생각하는 사람도 있다. 오늘날은 수학적식이 아니라 다른 곳에도 활용되는 경우가 많으며, L<sup>A</sup>TeX과 같은 형식이 보편적으로 쓰이고 있다.



〈그림 1〉 TeX logo

## 2 TeX이란?

오늘날 TeX이라는 용어는 흔히 몇 가지 의미로 쓰인다.

**조판 프로그래밍 언어** TeX은 일종의 컴퓨터 언어이다. 강력한 매크로 처리가 특징이며, 주로 하는 일은 가상의 평면에 어떤 object들(주로 박스로 이루어진 문자, 그림 등)을 배치하는 ‘조판(typesetting)’이다. 그러나 그 자체로도 이미 하나의 언어이기 때문에 내부적으로 계산, 제어 등이 가능하다. TeX을 이용하여 문서를 생성하는 것은 다르게 말하면 “문서를 프로그래밍하는 것”이라고도 할 수 있다.

**컴파일러** C 언어의 컴파일러가 예컨대 cc 또는 gcc와 같은 프로그램으로 존재하듯이 TeX 언어도 컴파일러가 있다. 이것이 tex이라는 이름의 프로그램이다. 이것은 대화

<sup>1</sup><http://en.wikipedia.org/wiki/TeX>

형 프로그램으로서 콘솔에서 실행하여 한 스텝씩 입력된 명령을 처리할 수 있다. 실제로는 미리 작성된 명령 집합을 파일로 만들어 `tex` 프로그램에 전달해주는 형태로 쓴다. 윈도에는 `tex.exe`라는 형태로 존재한다. `tex`의 입력은 콘솔 또는 파일이지만 출력은 `.dvi`라는 확장자를 갖는 파일로 이루어진다. `.dvi`는 조판 결과가 담겨 있다.<sup>2</sup>

**매크로** TeX으로 실제 조판을 하려면 TeX 언어의 세부까지 알아야 하고 객체에 대하여 하나하나 명령을 지시해야 한다. 이것은 너무 힘든 일이기 때문에 일종의 라이브러리로서 미리 정의된 매크로를 이용하여 조판 작업을 손쉽게 할 수 있게 하였다. 가장 간단한 매크로 집합은 D. Knuth 자신이 작성한 `plainTeX`이라는 매크로 집합이며, 가장 널리 쓰이는 것은 `LATeX`이라는 (원래 Leslie Lamport가 처음 작성했던) 매크로이다. 요즘은 TeX이라고 하면 `LATeX`을 가리키는 경우도 많다.

**TeX 실행 시스템** TeX을 실제 자신의 컴퓨터에서 실행하려면 방대한 보조 장치가 필요하다. 대표적인 것이 폰트인데 폰트가 없으면 만들어진 조판 결과를 출력할 수 없기 때문에 TeX 시스템의 필수적인 구성요소가 된다. 또한 `dvi`를 적절한 출력장치(예, 스크린)에 보여주거나 프린트하기 위해서는 그 목적에 사용될 유틸리티도 필요하다. 이 모두를 하나의 묶음으로 만들어 손쉽게 TeX을 설치하여 운영할 수 있게 해주는 일련의 엔진, 매크로, 폰트, 유틸리티의 집합을 TeX 실행 시스템 또는 TeX 배포판<sup>3</sup>이라고 한다. 가장 유명한 배포판으로 텍사용자그룹(TUG)이 매년 제작하여 배포하는 TeX Live가 있으며 윈도용의 MiKTeX이라는 프로그램도 일종의 텍 실행 시스템이다. 사용자는 이 시스템을 설치함으로써 TeX을 실제로 사용할 수 있게 되는 것이다.

이 가운데, 텍 실행 시스템의 네 가지 구성요소에 대하여 좀더 자세히 살펴본다.

## 2.1 엔진

엔진이라 함은 최종적으로 조판 자체를 담당하는 핵심 프로그램을 가리키는 것이다. 앞서 말한 컴파일러가 여기에 해당한다. D. Knuth 박사가 만든 `tex`이라는 프로그램이 대표적인 ‘엔진’ 프로그램이다.

최근 들어 TeX 엔진에 대한 다양한 개량이 이루어졌다. 그 중 괄목할 만한 것으로는 TeX을 직접 확장한 `ε-TeX`, 출력을 `dvi`가 아닌 `pdf` 포맷으로 직접 할 수 있고 `pdf` 특유의

<sup>2</sup>현재 가장 널리 쓰이는 TeX 엔진인 `pdfTeX`은 `dvi` 없이 `pdf` 파일을 곧바로 출력한다. 그래서 원래 `tex` 프로그램의 출력 형식이 `dvi`였던 사실을 잊고 있는 경우가 많다. 그러나 아직도 특별한 목적을 위해서 `dvi` 출력을 써야 하는 경우가 있다.

<sup>3</sup>TeX implementation

명령을 더 잘 처리하는 pdfTeX, 텍 고유의 폰트 처리 방식을 수정하여 시스템에 설치된 트루타입·오픈타입 등의 폰트를 자유롭게 사용할 수 있게 한 XeTeX, pdfTeX을 바탕으로 유니코드를 기반으로 하며 lua언어를 탑재하고 metapost 엔진까지 포함시켜 TeX 엔진을 대폭 확장한 차세대 TeX 엔진으로 불리는 LuaTeX 등이 그러하다. 이 가운데 서구어의 조판에서 가장 많이 쓰이는 것은 pdfTeX이라고 할 수 있으나 한글 문헌의 조판에는 폰트 사용의 편의성 때문에 XeTeX이 얼마 전부터 널리 쓰이고 있다.

## 2.2 매크로

plainTeX 과 L<sup>A</sup>TeX<sup>4</sup>은 대표적인 매크로 세트이다. 특히 L<sup>A</sup>TeX은 오늘날 과학 기술 문헌의 원고를 작성하거나 출판하는 데 ‘사실상의 표준’이 되어 있다. TeX이 가장 널리 쓰이는 수학 분야에서는 L<sup>A</sup>TeX에 수학적 표현을 강화한  $\mathcal{A}\mathcal{M}\mathcal{S}$ -L<sup>A</sup>TeX 이라는 매크로 세트를 많이 사용한다.

이밖에 사용자가 많지는 않으나 점점 늘어나고 있는 ConTeXt라는 매크로 집합도 있다.

## 2.3 폰트

폰트는 텍 실행 체계에서 가장 많은 양을 차지하는 부분이다. 원래 폰트는 TeX 시스템의 일부로 Knuth 박사 자신에 의하여 개발되었다. 그것이 Computer Modern 글꼴이라 불리는 것이며, 이 글꼴을 디자인하기 위하여 METAFONT라는 폰트 기술언어를 별도로 만들었던 것이다. 사자는 TeX을, 암사자는 METAFONT를 상징하는 로고이다. CM 글꼴은 지금도 TeX으로 작성하는 문서의 기본 글꼴이다.<sup>5</sup> 그림 2는 Knuth 교수의 저서인 *The TeXbook*에 사용된 사자 그림이다.

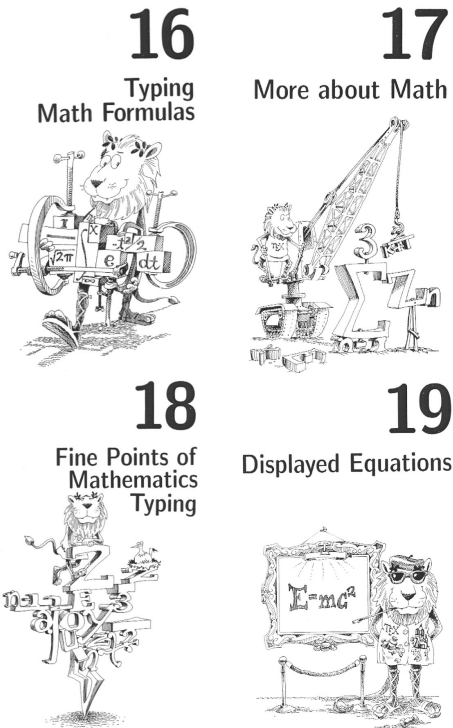
지금은 CM 이외의 PostScript, Truetype, Opentype 글꼴들을 사용할 수 있게 되었으며 심지어 운영체제에 설치된 폰트까지도 활용할 수 있을 정도가 되었다. 그렇지만 pdfTeX과 같은 legacy 텍 엔진<sup>6</sup>은 시스템에 설치된 폰트를 사용하지 않으며 별도의 폰트 사용 인터페이스를 가지고 있기 때문에 폰트를 수작업으로 설치하거나 설정하는 것이 쉬운 일이 아니었다. 대부분의 텍 실행 체계는 설치시 이 까다로운 부분을 대부분 자동 설정해주어서 사용자에게 편의를 도모하고 있다.

한글의 경우, ‘은 글꼴’이라는 공개 글꼴(GNU GPL)을 기본 글꼴로 채택하고 있다. 이 글꼴은 은광희 박사가 H<sup>A</sup>TeX과 함께 제작한 포스트스크립트 글꼴과, 박원규 선생이

<sup>4</sup>“레이텍” 또는 “라텍”이라고 읽는다. “라텍스”는 절대로 금물.

<sup>5</sup>엄밀히 말하면 CM 자체라기보다 CM으로부터 파생된 글꼴들이라고 하여야 한다. 가장 유명한 것으로 Latin Modern 글꼴이 있는데 기본적으로 CM으로부터 만들어졌다.

<sup>6</sup>최근 XeTeX과 LuaTeX의 등장으로 원본 TeX이나 pdfTeX을 이렇게 부르게 되었다.



<그림 2> TeX Lion

이로부터 트루타입 형식으로 변환한 “은 글꼴”로 이루어져 있다. 트루타입 폰트는 운영 체제에 설치하여 다른 목적으로도 사용할 수 있다. 한 벌의 자유 글꼴을 가지고 있다는 것만으로도 우리는 KTUG의 기여를 매우 자랑스럽게 생각한다.

폰트 문제에 있어 최근의 텍 엔진들인 XeTeX, LuaTeX 들은 전통적인 텍 폰트뿐 아니라 오픈타입 또는 트루타입 형식의 시스템 폰트를 자유롭게 활용할 수 있게 되었다. 텍 발전의 역사에 있어 이 부분은 획기적이라고 할 만한 변화를 가져왔다고 할 수 있다. 한글 글꼴로 한글과컴퓨터가 공개한 ‘함초롬체’ 글꼴을 유니코드 환경에 걸맞은 (옛한글 표현까지 자유로운) ‘함초롬체 LVT’로 수정한 것도 KTUG의 중요한 기여이다.

폰트 문제가 해결되면서 얻게 된 중요한 기여로서, 1990년대 초중반부터 2000년대 말 까지 “텍으로는 완성형(EUC-KR) 글자밖에 찍을 수 없다”는 생각을 불식하게 된 것이다. 실제 최근의 ko.TeX으로는 유니코드로 찍을 수 있는 모든 글자, 즉 현대한글 음절문자, 자모영역 글자, 심지어 한양PUA 옛글자까지 자유롭게 식자할 수 있게 되었다. 고문헌 등의 식자와 조판에 있어서도 텍이 한 표준이 될 수 있으리라 생각한다.

```

\documentclass{article}
\usepackage{kotex}
\begin{document}
\title{제목}
\author{아무개}
\maketitle
\section{첫번째 소절}
안녕하세요, 한글!
\end{document}

```

〈그림 3〉 helloworld.tex

## 2.4 유틸리티

종래 TeX의 표준 출력이었던 dvi<sup>7</sup>는 그 자체로서는 출력에 필요한 정보만을 가지고 있으므로 어떤 식으로든 변환을 거쳐야 인쇄 또는 화면보기가 가능하였다. 이 일을 해주는 유틸리티를 dviware라고 하는데, 대표적인 것이 포스트스크립트 출력기용 코드(또는 파일)로 변환하는 dvips가 있으며, 화면보기를 위한 xdvi, yap, dviout 등이 있다. 또한 pdf로 변환하는 데 사용되는 dvipdfmx라는, KTUG의 조진환 박사(현 수원대 교수)가 제작한 유틸리티도 있다. XeTeX은 사용자가 느끼기에는 바로 pdf를 생성하는 것 같지만 실은 xdv라고 하는 dvi 확장 포맷으로 먼저 컴파일을 하고 그 결과를 xdvipdfmx라는 dviware가 pdf로 변환해주는 일을 일괄로 처리한다. xdvipdfmx는 dvipdfmx를 발전시킨 것이다. pdfTeX이나 LuaTeX은 직접 pdf를 생성할 수 있기 때문에 이 때는 엔진 자체가 dviware의 역할을 겸하고 있는 셈이다.

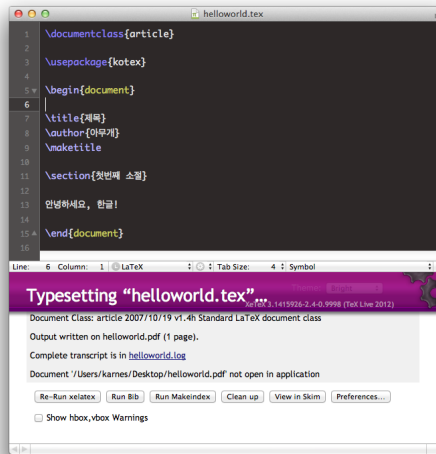
이밖에도 그림 변환을 위한 것, 출력물을 조정하기 위한 것 등 텍으로 문서를 작성하는데 필요한 여러 가지 유틸리티들이 포함되어 사용자의 작업을 매우 편리하게 만들고 있는 것이다.

## 3 TeX 사용의 몇 가지 측면

### 3.1 기본적인 L<sup>A</sup>TeX 문서 원본의 구조

L<sup>A</sup>TeX 문서는 플레인 텍스트를 편집하는 에디터로, 그림 3과 같이 입력하는 것으로 시작한다.

<sup>7</sup>이 파일 확장명의 의미는 “장치독립”이라는 것이다. 특히 어떤 시스템에서도 동일한 결과를 얻으려는 텍의 철학이 잘 반영된 포맷이다.

〈그림 4〉 L<sup>A</sup>T<sub>E</sub>X 편집

문서의 편집에는 어떤 에디터를 사용해도 상관없다. 그림 4는 매킨토시 컴퓨터에서 `textmate`라는 에디터로 위의 텍 파일을 편집하고 있는 그림이다.

이 첫 문서에서 보는 바와 같이 일본(소스) 문서에는 두 가지 요소가 포함되어 있다. 하나는 콘텐츠를 구성하는 텍스트 자체이고, 다른 하나는 그 콘텐츠를 실제로 조판하기 위하여 필요한 조판 지시 마크업이다. 이 두 가지 요소를 적절히 사용하여 작성한 원본 파일을 `tex` 또는 `latex`으로 하여금 처리하게 하면 결과물로서 `dvi`나 `pdf`를 얻어낼 수 있는 것이다. L<sup>A</sup>T<sub>E</sub>X이란 조판 지시 마크업을 미리 정의해두어서(예컨대 `\documentclass` 나 `\section{...}`과 같이) 사용자가 좀더 직관적으로 쉽게 조판 지시 마크업을 활용할 수 있도록 한 매크로 라이브러리라고 생각할 수 있다.

이 과정은 프로그래밍과 대단히 유사하다. 즉 미리 약속된 일련의 명령어를 나열하여 이것을 컴파일러에게 보내면 컴파일러가 `object` 코드를 생성하는 것과 동일하게, 미리 약속된 일련의 매크로 명령을 이용하여 문서를 작성하고 이것을 `tex` 엔진에게 보내면 `tex`이 이로부터 어셈블된 문서를 생성하는 것이다. `tex` 엔진을 (비유적인 표현이기는 하지만) “컴파일러”라고 부르는 이유가 여기에 있다.

또한 이러한 “마크업” 문서 작성은 HTML이나 XML 문서와 그 구조가 거의 동일하다고 볼 수도 있다. XML이 태그를 붙여 문서의 형식과 구조를 기술하는데, T<sub>E</sub>X은 명령어를 마크-업하여 조판 상태를 제어한다. 그 목적이나 용도는 다르지만 마크-업 문서 작성이 낮은 것은 아니다.

$$\prod_{j \geq 0} \left( \sum_{k \geq 0} a_{jk} z^k \right) = \sum_{k \geq 0} z^n \left( \sum_{\substack{k_0, k_1, \dots \geq 0 \\ k_0 + k_1 + \dots = n}} a_0 k_0 a_1 k_1 \dots \right)$$

〈그림 5〉 수식 예제

### 3.2 TeX으로 문서작성하는 것의 효용

TeX 또는 LaTeX으로 문서를 작성(프로그래밍)하는 것은 몇 가지 장점이 있다.<sup>8</sup>

- (1) 문서 요소를 자동화할 수 있다. 예컨대 장절의 번호매기기, 색인 만들기, 상호참조 등의 요소를 간단한 코드를 통하여 완전하게 자동화할 수 있다.
- (2) 미리 정의된 매크로 세트를 이용하여 공동작업자들 간에 문서 포맷의 일관성을 유지할 수 있다. 특히 학술지 등을 편집할 때는 여러 기고자들에게 동일한 포맷을 요구하게 되는데, 이 목적에 매우 충실하며, 기고자는 포맷에 대해서는 신경쓸 것 없이 오직 내용만을 작성하면 되도록 한다.
- (3) 아름답고 표준적인 수학적식을 조판할 수 있다(그림 5 참고). 이것은 대단히 중요한 장점으로, 실제 미국수학회 등에서는 TeX 수식이 수식 표현의 표준이다. TeX 언어는 수식의 스페이싱이나 폰트 등을 매우 잘 제어할 수 있어, 전문적인 수식을 간단히 출력할 수 있는 것이다.
- (4) 견고한 이식성. 문서를 직접 자신이 디자인하고 출력하는 데 있어 자신의 아이디어를 충분히 구현할 수 있으며 그것이 견고한 코드로 유지된다는 것이다. 다시 말하면, 예컨대 HWP 97에서 작성한 문서가 나중 버전에서 확실히 열릴 것이라는 보장이 없는 것과는 달리, 원본 소스는 언제나 플레인 텍스트로 유지되므로 텍 시스템이 존재하는 한 그 결과는 항상 동일하게 출력할 수 있을 것이며, 언제라도 그 내용을 확인할 수 있을 것이다.

### 3.3 TeX의 다른 활용 가능성

TeX은 자유소프트웨어이며 프로그래밍 언어이기 때문에 “책을 만드는” 본래의 목적은 물론이고, 확장된 사용 가능성이 대단히 크다.

한 예를 들자면, 위키나 웹 페이지에서 수식을 표현해주는 데 TeX이 사용되는 것을 들 수 있다. 많은 위키 엔진에서 웹 페이지에 TeX 수식을 표현해주는 add-on을 가지고 있다.

<sup>8</sup>Allin Cottrell, 김강수 (옮김), “워드 프로세서, 명칭하고 비효율적인 도구”, <http://people.ktug.or.kr/~karnes/wp-ko/> 참조.



또한, MySQL 등의 데이터베이스와 연동하여 다양한 작업을, 워드 프로세서 등으로는 상상하기 힘든 작업을 하는 경우도 있다. 그 한 예로 KTUG Conference에서 김병룡 박사에 의해 소개된 사전 조판을 들 수 있는데, 사전의 항목들을 미리 데이터베이스에 입력해두고 그것을 여러 방식으로 출력하는 것이 가능함을 보여주었다.<sup>9</sup>

이런 특징을 이용하여, 블로그 페이지를 자신만의 책으로 출판하는 일은 이미 국내에서 성공적으로 행하여진 바가 있다고 한다. 이외에도 다른 프로그래밍 언어와 결합하여 지금까지 경험하지 못한 새로운 결과를 얻게 되는 데 유용한 고품위 출력 엔진으로 TeX이 사용될 수 있을 것이다.

#### 4 KTUG과 TeX에서의 한글 사용

TeX은 처음에 7(8)비트 체계로서, 영문자를 식자하는 프로그램으로 시작하였다. 그러나 텍이 가진 여러 장점을 다른 언어로도 활용하고자 하여 현재는 거의 대부분의 언어를 텍이 식자할 수 있게 되었다.

한글 사용 노력은 1980년대 후반부터 시작되었다. 그런 노력의 한 결실이 KAIST를 중심으로 개발되었던 H<sub>A</sub>TeX과 한TeX이었다. 그러나 폰트와 관련된 문제로 완전한 공개가 이루어지지 못하여, 공개 폰트를 완전히 새로 제작하고 매크로 패키지를 구성한 은광희 박사의 H<sub>A</sub>TeX이 사실상의 표준 위치를 고수하다가 2007년을 전후하여 ko.TeX으로 발전하였다. ko.TeX은 H<sub>A</sub>TeX과 당시 발전하고 있던 유니코드 한글 식자 시스템을 결합하여 출발하였으며 현재는 XeTeX과 LuaTeX을 위한 한글 식자 시스템을 개발하여 가고 있다.<sup>10</sup> 이 일이 이루어진 배경에는 KTUG의 활동이 뒷받침되어 있다.

KTUG이 2001년 활동을 시작한 이래, TeX과 그 관련 프로그램에서 한글을 구현하고 고품위의 pdf를 제작하는 기술을 집중적으로 발전시켜왔다. 그 결과, dvipdfmx라는 dviware 유틸리티<sup>11</sup>와, Hangul-ucs라는 유니코드 한글 L<sub>A</sub>TeX 시스템<sup>12</sup>을 성공적으로 개발하여 이를 바탕으로 ko.TeX을 탄생시켰으며, 이제는 한글 문헌을 식자 조판하는 데 있어, 그 출력품위가 어떤 워드 프로세서나 레이아웃 프로세서에 비교해서도 뛰어나다고 할 수 있을 정도가 되었다. 또한 한글화의 범위도 종래의 L<sub>A</sub>TeX에 국한되지 않고 ConTeXt와 XeTeX, LuaTeX까지 그 영역을 넓혀가고 있다.

이를 통하여, 비단 수학 등의 과학 문헌 조판을 넘어서서, 인문학과 사회과학, 공학, 일반문헌의 조판 등으로 그 가능성을 확장하고 있는 것이다.

<sup>9</sup>김병룡, KTUG 5주년 학술발표회, “TeX 과 DB와의 만남.”

<sup>10</sup>조진환 (2007), “한글 라텍이 걸어온 길과 ko.TeX,” AJT, vol.1, no.2. <http://ajt.ktug.kr/2007/0102chof.pdf>. 현재 ko.TeX은 핵심적인 개발을 김도현이 맡고 있으며 EUC-KR을 위한 지원과 oblvioir 클래스의 개발을 김강수가 담당하고 있다.

<sup>11</sup>조진환(수원대 교수) 및 S. Hirata 개발

<sup>12</sup>김도현(동국대 교수) 및 김강수(한국텍학회 부회장) 개발

## 5 결어

TeX 자체가 빠르게 발전하고 있는 프로그램이며, 앞으로 개발될 TeX은 이전과는 비교할 수 없을 정도로 확장된 기능을 가지게 될 것이다.

비록 처음에 익숙해지는 데 약간의 노력이 필요하여 대중성에 흠결이 있어 보이는 것이 사실이라 하더라도 TeX 시스템 자체는 이미 그 조판 능력과 견고성을 30년의 세월 동안 충분히 입증하였다. 출현한 지는 근 30년에 이르게 되었으나 지금도 새로운 프로그램이며 언어이다.<sup>13</sup>

---

<sup>13</sup>김강수, “TeX 과 그 언저리” 및 같은 저자의 “H<sub>2</sub>TeX 이후 한글 L<sub>2</sub>TeX 의 발전”을 참고.