

TEX이 만들어지기 까지

작은나무 번역

2006년 8월 22일

이 글은 TEX개발의 전체적인 역사를 기록한 *The Errors of TEX*이라는 Knuth의 글에서 TEX 개발에 대한 과정을 소개하는 부분만을 번역한 것입니다. 특히 이 부분은 컴퓨터 프로그래밍에 대한 지식이 없어도 쉽게 읽을 수 있는 부분이며, TEX이 어떠한 과정을 통해서 만들어 졌는지를, 그리고 저자 Knuth는 얼마나 많은 시행착오를 거치며 그 멋진 일을 해내었는지를 알게되는 재미있는 글입니다.

개발 초기

TEX이 그 잉태를 시작한 날은 1977년 2월 1일이었는데, 그날은 처음으로 고품질 조판 기계의 조판물을 본 날이기도 합니다. 그 조판물은 품질이 매우 우수했고 그것을 만든 방식은 순전히 디지털 방식으로 만든 것이라고 들었는데, 실제 기계로 조판한 것과 차이를 느끼지 못했습니다. 그래서 나는 그 복잡하고 정교한 조판의 핵심 부분도 컴퓨터로 간단하게 만들 수 있다는 것을 깨달았습니다. 전산학자로서 그러한 조판 프로그램을 더 발전시켜서 현재보다 더 나은 품질의 조판물을 만들어 낼 수 있다는 생각을 하지 않을 수 없었습니다. 그래서 2월 8일의 내 일기장에는 이미 그때부터 스탠포드 대학의 인공지능 연구소에서 근무하는 연구원들과 새로운 조판 소프트웨어의 가능성에 대해서 논의한 내용이 적혀 있습니다. 2월 13일, 나는 곧 다가오는 안식년을 남미에서 보내기로 한 계획을 수정했습니다. 남미의 이국적인 장소를 여행하면서 *The Art of Computer Programming*의 제4권을 집필을 하는 것 대신에 스탠포드에 남아서 디지털 조판에 대해서 연구를하기로 결심한 것입니다.

TEX의 디자인 고안하기 시작한 것은 1977년 5월 5일 입니다. 그로부터 일주일 후, 나는 내 생각에 매우 완벽한 디자인 초안을 작성했고, 그 내용을

타이핑해 입력해 넣느라 새벽 5시까지 컴퓨터 앞에 앉아 있었습니다. 그러고나니, 조판에 관해서는 거의 모든 것들이 명확해져서 무엇을 어떻게 해야 할지를 정확히 알게되어, 조판에 관한 일은 잠시 미뤄두고, 대신에 지금의 METAFONT로 알려진 프로그램의 초기 모델을 작성하느라 45일 정도를 보냈습니다. 그리하여 6월 28일에 나는 25개의 영문 알파벳 소문자를 여러가지 폰트에 대해서 만들 수 있게 되었고, 그것들은 그 당시 내 기준으로는 꽤나 괜찮게 보였습니다. 그로부터 3일 후에 나는 마지막 26번째 글자를 어떻게 다루어야 하는지를 알게 되었는데, 그 글자는 기존 방법과는 다른 새로운 아이디어를 필요로 했습니다.

7월 3일부터 다시 T_EX에 대한 연구를 재개하게 되었습니다. 그동안 여러 명이 내가 작성한 초안을 읽고 사려깊은 조언과 충고를 해주었으며, 나는 그 후 2주간 더 연구해서 완전히 새롭게 개정된 디자인을 마련할 수 있었습니다. (이 기간동안 나는 영어의 hyphenation에 대한 알고리즘을 개발하기 위해서 거의 이틀 동안 사전과 함께 살기도 했습니다.) 이러한 연구의 결과로 내 생각에는 완벽한 조판 언어의 명세서를 얻을 수 있었고, 그래서 나머지 일은 그 해 여름 방학 동안 나의 연구 조교로 있던 대학원생 두 명(Frank Lian과 Michael Plass)에게 맡기었습니다. 그 두 명이 해야 할 일은 내가 중국을 방문하느라 잠시 떠나 있는 동안에 T_EX을 구현해 내는 것이었습니다. 8월 25일, 나는 중국에서 돌아왔고, 그들과 하루 만남을 가진 후에 다시 3주간의 여행을 떠나게 되었습니다. 9월 14일, 나는 그 여행에서 돌아왔고, 그들은 T_EX프로그램의 프로토타입을 코딩해서 나에게 보여주었습니다! 그들은 전체 시스템으로 보면 단지 15% 정도만을 구현했으며, 그들이 사용하기 위해 만들어낸 자료구조는 나머지 85%에 적용하기에는 범용적인 면이나 효율적인 면에서 너무나 부족한 점들이 많았습니다. 하지만 그 15%에 한정하면, 현명하게 구현해 내었고, 그로 인해서 작은 테스트 프로그램을 그 프로토타입으로 처음부터 끝까지 돌려볼 수 있게되어, 온전한 시스템이 어떠한 모양을 이루어야 하는지에 대해 상상하는데 도움이 되기도 하였습니다.

Liang과 Plass은 여름 방학이 끝나고 다시 학업으로 돌아가야 했고, 때마침 나는 안식년을 맞이하게 되어서 9월 16일에 (적어도 내 생각으로는) “T_EX의 최종 버전”을 직접 스스로 코딩하기 시작했고, 그러면서 나는 그들이 여름 방학동안 한 작업이 얼마나 대단한 결과물이었다는 것을 알게되었습니다. 나는 내가 작성한 T_EX의 명세서가 꽤나 완벽하다고 생각했는데 직접 코딩을 해나가는 도중에 그 명세서가 얼마나 부족한 것인지 알게되었습니다. 코딩을 하면서 거의 매 15분마다 새롭게 구현해야 할 기능들이 나타나는 것이었습

니다. 만약에 내가 그 두 대학원생의 입장이었다면, 즉 시스템의 고안자는 어디론가 떠나가 있어 연락을 취할 수 없는 상태에서 그가 고안해낸 시스템을 실제로 구현해내어야 하는 상황이었다면, 나는 아마 포기했었을 것입니다. 구현해 나가는 때 순간마다 시스템에 있어서 중요한 결정을 내려야 하는 곳이 한 두 곳이 아니었기 때문입니다.

이것으로 인해 내가 T_EX 프로젝트를 진행하면서 얻은 최초의 큰 교훈은 다음과 같습니다. 시스템을 고안해 낸 사람은 반드시 그것의 실제 구현에 완전히 참가해야 한다는 것입니다. 내가 중국으로 떠나지 않아서 나의 학생들이 나와 의논할 수 있는 상황이었다면, 그들은 아마도 매우 중요한 문제를 가지고 한 시가 멀다하고 나를 찾아왔을 것입니다. 그들과 함께 각각의 특별한 문제를 가지고 한 시간 이상씩 의논했을 것이고, 아마도 수 백번의 회의를 가졌을 것입니다. 나는 그동안 내가 알고있던 다른 프로젝트가 실패한 이유를 알았습니다. 그 실패한 프로젝트에서 그 시스템의 고안자는 실제로 코딩을 하며 구현하고자 하지 않았던 것입니다.

10월 14일, 나는 수식 조판 부분과 T_EX의 내부 표현을 출력 장치를 위한 코드로 변화하는 부분을 제외한 대부분의 코딩을 마쳤습니다. 이 시점에서 나는 유럽으로 3주 동안 여행을 떠나야 했습니다. 이번 유럽 여행은 아주 오래전에 계획되었던 것이라, 최근에 연구하고 있는 디지털 조판 시스템과는 무관한 주제에 관한 여행이었습니다. 그러나 나는 독일의 오브월팍과 노르웨이의 오슬로 등지에서 수학자들과 곡선을 그리는 재미있는 문제에 대해서 이야기를 나눌 수 있었습니다. 더구나 나는 영국의 레드힐에 위치한 Monotype Corporation의 본사를 방문할 기회를 갖기도 했습니다.

그 유럽 여행에서 돌아온 나는 Computer Modern type의 최초 고안에 포함되어 있는 숫자, 알파벳 대문자와 구뚝점에 대한 폰트를 만들어내느라 11월을 보냈습니다. 나는 일전에 끝마친 소문자 뿐만 아니라 모든 것을 포함하는 완전한 폰트를 만들어 내야 했는데, 이는 American Mathematical Society에서 내가 그동안 해낸 폰트 작업에 대해서 강의를 해달라고 요청을 받았기 때문입니다. 그 강의에서 나는 알파벳 소문자들만을 보여주고 싶지는 않았습니다. 나는 12월 동안 그 AMS 강의 준비를 하였고 다음해 1월에 폰트에 대한 나의 작업 결과에 대한 강의를 하였습니다. 그래서 1월 14일이 되셔야 비로소 다시 T_EX의 코딩을 다시 시작할 수 있었습니다. 그 일은 아주 순조롭게 잘 진행되어서 1978년 2월 10일에 나의 일기장에는 다음과 같이 적혀있습니다.

그동안의 미해결 문제들을 포함해서 모든 T_EX 프로그램들을 끝
마쳤고, 문법 에러 없이 모든 컴파일을 마쳤다 (4 a.m.).

TeX은 1970년 이후로 내가 작성한 최초의 실로 거대한 프로그램 이었습니다. 즉 “구조적 프로그램”의 관점에서 보아 나의 첫 프로그램인 것입니다. 나는 70년대 초에 Dijkstra, Hoare, Dahl과 그외 많은 분들로 부터 구조적 프로그래밍 기법을 배웠습니다. 나는 구조적 프로그램 기법이 내가 작성하는 코드에 대한 정확도를 높여주리라는 확신에 차 있었지만, 그것을 시험해 볼 기회가 없었고, 코드들은 아직까지는 나의 연습장에만 존재해있었습니다. 그래서 이 TeX 프로젝트가 구조적 프로그래밍 기법에 대한 적절한 테스트 기회라고 생각되어서 TeX 코드를 작성해 나가면서 시시때때로 작성하고 있는 코드에 대해서 디버그나 테스트를 하지 않고, 구조적 기법으로 모든 코드를 작성한 후에 과연 내가 얼마나 정확한 코드를 작성해 내었는지를 판단하기 위해서 끝까지 기다렸습니다. 이 같은 나의 판단 덕분에 프로그램을 작성해 나가는데 많은 시간을 절약 할 수 있었습니다. 왜냐하면, 프로그램이 올바르게 동작하는지를 테스트 하기 위해서 쓸데없는 돌들을 만들어 내지 않아도 되었기 때문입니다. 물론 구조적 프로그래밍 기법만 믿고, 9월 부터 작성하기 시작한 코드가 중간에 테스트도 하지 않은 채로 모든 코드를 완성해 나가는데 있어서 이 프로그램을 거의 완성할 즈음인 1월에 나는 슬슬 과연 이 프로그램이 제대로 동작할 것인가에 대한 불안한 마음이 들기도 했습니다. 그러나 이 모든 나의 결심 때문에 프로그램을 빨리 테스트 해보고자, 프로그램을 좀 더 빨리 작성하게되는 동기부여가 되기도 했습니다.

2월 10일, 드디어 그동안 작성한 TeX 프로그램 코드가 모두 완성되어 컴파일까지 마쳐서 테스트할 만반의 준비가 다 되었습니다. 하지만 나에겐 무언가에게 쫓겨 서둘러야 할 이유가 전혀 없었습니다. 나는 내 프로그램 코드를 읽기 편하게 작성하였고 또한 “비공식적으로나마” 프로그램이 올바르게 동작한다고 증명되었기 때문에, 나는 그후 한 달을 이탤릭체, 희랍어, 첨자, 기호 폰트들을 만들면서 보냈습니다. 나의 프로그램, TeX을 테스트하기 위해서는 그 모든 폰트들을 필요로 했고, 그 모든 것들이 갖추어 지지 않은 상태에서 테스트를 진행하고 싶지 않았기 때문입니다. 이 과정에서 또다시 나는 단순히 테스트만을 위한 코드를 작성하지 않아서 시간을 절약할 수 있다는 것을 알게되었습니다. 이처럼 구조적 프로그래밍 기법은 나에게 모든 코드를 완성해 나가는데 있어서 과연 내가 올바른 프로그램을 만들고 있는지를 테스트하기 위한 코드를 작성하지 않아도 될 만큼의 용기를 불어 넣어 주었습니다. 3월 8일에 큰 기호들의 폰트를 모두 만들어 내었고, 3월 9일 일기장에는 다음과 같은 행복한 글이 실리게 되었습니다.

그동안의 모든 수정 내용을 TeX 프로그램에 반영해서 컴파일을

끝마쳤다—드디어 내일부터 디버깅 시작이다!

TeX에 대한 모든 에러나 버그를 기록한 나의 로그 목록이 기록된 책은 그 다음 날 부터인 3월 10일부터 기록되었습니다. 3월 29일, TeX이 정상적으로 작동하기 시작했다고 스스로 판단을 내리기에 이르렀습니다.

... (에러 복구 루틴을 제외하고는)—드디어 축하할 시간이 되었다!

나는 폰트들을 좀 더 세밀하게 다듬고, TeX 매뉴얼에 대한 초안을 구상하기 시작했습니다. 그리고나서 스탠포드가 견본 조판 기계를 구입하려고하는 일리노이에 있는 Alphatype Corporation을 방문하느라 몇 일을 보냈습니다. 4월 11일부터 5월 11일까지는 *The Art of Computer Programming*의 제2권인 *Seminumerical Algorithms*를 몇 군데 수정해서 개정판을 만드느라 조판 연구에서 잠시 떠나 있었습니다. 나는 그동안 내가 만든 TeX 프로그램을 그 개정판 작업에 투입해 보고자 했고, 그 개정판은 최초의 실질적인 TeX 적용 사례가 되었습니다. 그리하여 5월 14일 나는 다시 TeX을 작동해 보았고, 제2권의 iv쪽부터 8쪽까지가 그 시험 대상으로써 제록스 그래픽스 프린터로 출력되었습니다. 이날이 5월 15일 이었습니다.

그후 몇 주 동안 나의 작업은 나에게 딱 맞는 일들이었습니다. 제2권의 원고를 준비하면서 나는 스스로 내 TeX의 사용자가 되었습니다. 이것은 매우 값진 경험이었습니다. 그즈음이 9월 24일, 나의 안식년도 끝이 났습니다. 나는 700쪽이나 되는 개정판의 441쪽까지 타이핑을 마쳤습니다. 그 원고를 TeX으로 준비하면서 나는 TeX의 개선 사항을 꾸준히 기록해 두었고, 그 기간 중에 한달간 나의 가족들과 여행을 할 기회가 있었는데, 그때는 잠시 잊으려고 노력했습니다.(그런데 사실 그 휴가 기간에도 돌아다니면서 눈에 보이는 모든 폰트들을 유심히 살펴보고, 그런 모양의 폰트는 어떻게 하면 컴퓨터로 그릴 수 있을 것인가를 계속 생각했습니다. 하루는 아침에 그랜드캐년의 trail에 앉아서 METAFONT를 위한 대수적 표기법을 고안하면서 보냈습니다. 내가 만들어낸 폰트들은 프리미티브 매크로 언어로 작성되고 난 다음에 컴파일된 것들입니다. 인터프리터된 것들이 아닙니다.) 그러도 또한 그해 여름의 3주를 최초의 TeX 매뉴얼을 만드는데 보냈습니다.

비록 나의 안식년은 끝나지만, 가을 학기의 강의 중 비는 시간을 이용해서 조판 연구를 계속 진행했습니다. 제2권은 11월 15일 아침에 끝마칠 수 있었습니다. 11월 17일, 나는 METAFONT를 작성하기 시작했습니다. 1978년 12월 31일 나의 일기에는 다음과 같이 쓰여져 있습니다.

METAFONT 작성을 모두 끝마쳤다. 축하할 일이다. 새해의 전날
(11:59 p.m.).

1978년 8월부터 다른 사람들이 T_EX을 사용하기 시작했고, 놀라운 속도로 많은 사람들에게 전파되어서 놀라기도 했습니다. 1979년 1월부터 3월까지 여가시간은 T_EX을 파스칼 프로그래밍 언어로 작성할 수 있을 것인가에 대해서 생각하면서 보냈습니다. (오리지널 프로그램은 SAIL 이라는 몇몇 컴퓨터에서만 돌아가는 프로그래밍 언어로 작성했습니다.) 이 기간동안 나는 파스칼 프로그램의 소스 코드를 보기 좋게 조판하는 실험을 시작했습니다. 나는 BLAISE 라는 프로그램을 작성했는데, 이 프로그램은 파스칼 소스 코드를 T_EX 코드로 변환해 주는 프로그램입니다. BLAISE는 곧바로 DOC이라는 구조적 문서화를 위한 시스템으로 발전되었고, DOC 프로그램은 1979년 3월 31일에 완성되었습니다. DOC 형태로 작성된 프로그램은 파스칼 혹은 T_EX 코드로 변환이 가능했습니다. Luis Trabb Pardo와 Ignacio Zabala는 DOC을 사용해서 파스칼 코드로 작성된 T_EX버전을 만들어 냈는데, 이는 매우 이식성이 높은 코드였고, 1980년 4월에 완성되었습니다.

이 시기에 나는 또다른 큰 교훈을 얻었습니다. 소프트웨어를 작성하는 것이 책을 집필하는 것보다 훨씬 더 힘들다는 것입니다. 나는 학생들을 가르치는 일과 조판에 대해서 연구하는 일을 동시에 진행할 수 없었습니다. 그래서 1979년 봄학기에 강의를 맡지 않겠다고 요청을 했습니다. 3월 22일 나의 일기장에는 다음과 같이 적혀있습니다.

강의를 진행해야 한다는 무거운 짐을 훌훌 털어버렸다. 산적해 있는 T_EX에 관한 연구를 재개할 때이다.

(결국은 나의 안식년과 다음 안식년 사이의 기간동안의 21개의 학기동안 13학기 만을 강의를 하였습니다. 1983년에 대학원생들을 지도하는 것은 계속 하였으나 강의는 맡지 않았습니다. 그 1983년은 T_EX과 METAFONT에 대한 연구가 절정에 달해 있을 때 였습니다. 나는 82, 84, 85년도에 세달 동안 강의를 맡지 않았습니다. 나는 진심으로 학생들을 가르치는 것을 좋아합니다. 하지만 그 모든 의무들을 수행해 나가면서 T_EX 프로젝트를 완성시킬 길이 보이지 않았습니다.)

1979년 4월 1일, METAFONT로 돌아왔는데, 그때 METAFONT는 이미 작성되었고, 디버그는 하지 않은 상태였습니다. 4월 28일, METAFONT가 작동하기 시작했습니다. 그리고나서 Alphatype 장비에 대한 소프트웨어를 고안하기 시작했고 이 일은 세달이 걸렸습니다. 그해 여름에 METAFONT 매뉴얼을

TeX으로 썼는데, 그러면서 TeX에 대한 경험을 한층 더 쌓을 수 있었습니다. 역시 그해 여름에 TeX은 American Mathematical Society로 부터 중요한 자극을 받았습니다. Barbara Beeton과 Michael Spivak을 포함해서 여러 명이 스탠포드에서 TeX 매크로를 개발하는 기회를 갖기도 했는데, AMS의 사람들은 몇 가지 중요한 TeX을 적용할 거리들을 소개 해주었는데, 그 중에 *Mathematical Reviews*의 인덱스를 생성하는 작업이 있었습니다. 그 작업은 TeX을 한계까지 몰고 갈 정도로 힘든 작업이었고, 이는 결과적으로 TeX의 성능을 향상시킬 수있는 계기가 되었습니다.

개발 후기

1979년 8월 14일, 나는 TeX이 근본적으로 완전하고 매우 안정된 프로그램이라고 느꼈습니다. 나는 산타크루즈에 있는 Western Institute for Computer Science 소속의 100여 명의 참가자들을 대상으로 TeX 프로그램의 개발과 디버깅에 대한 나의 경험을 강의했습니다. 그 당시 프로그램의 에러와 버그들의 목록을 담은 나의 로그 책은 420개의 에러 항목을 가지고 있었습니다. 이 갯수는 그다지 크지 않은 수로 최종 목록은 그 두배입니다! 나는 이러한 에러 목록들을 기록함으로써 많은 것을 배울 수 있었습니다. 나는 그 강의에서 오후 7시 반부터 9시 반까지 강의 할 정도로 열정적 이었습니다. (청중들도 마찬가지로 열정적이어서 밤 11시 반까지 계속해서 질문을 해대었습니다. 그래서 그때 TeX의 에러에 관한 논문을 작성하기로 마음먹어서 지금 이 글을 쓰고 있는 것입니다.)

나는 1979년의 12월과 1980년의 1월을 Computer Modern에 온전히 바쳤습니다. 왜냐하면 METAFONT로 다시 작성해야 했기 때문입니다. 그때 전산학이 엄청난 속도로 발전하고 있어서 또다시 제2권을 개정할 필요가 있었습니다. 나의 안식년동안 그것을 마무리 하고 싶었는데, 결국은 1980년 7월 29일 새벽 2시에 끝났는데, 이는 계획보다 2년이 늦은 것이었습니다. 1980년 후반에 TeX과 METAFONT에서 사용한 새로운 아이디어에 대한 논문들을 썼습니다.

그러나 TeX에 관한 나의 연구는 결코 끝나지 않았습니다. 1980년 2월 22일, 미국 전역에 걸쳐서 대략 50여명이 스탠포드에 모여 TeX 사용자 모임 (TUG)을 설립했습니다. 나는 거기서 몇 가지면에서 기존의 TeX을 다소 불편한 방식으로 바꿔도 괜찮겠냐고 물어보았습니다. 그런데 그렇게 변경하고 나면, 기존의 매뉴얼과 그들이 예전 방식으로 작성한 모든 파일들이 쓸모가 없어지게 되는 것이었습니다. 그런데, 그들 모두는 나의 계획에 조금도 반대

하지 않았습니다! 곧 TUG은 Richard Palais 회장의 지도 하에서 드라마틱하게 성장해 나갔으며, 결국은 국제적으로 발전하게 되었습니다. 내가 이렇게 열정적인 사람들을 실망시키면서까지 *The Art of Computer Programming*의 다음 권들을 집필하기 위해서 T_EX을 현재 상태로 방치해서는 안된다는 것을 깨달았습니다.

나는 좀 더 나은 “endgame 전략”을 실행해야 할 필요를 느꼈습니다. 그러자 무엇을 해야하는지가 명확해졌습니다. T_EX과 METAFONT의 오리지널 버전은 역사 속으로 사라져야 한다는 것입니다. 그 프로그램들은 조판 시스템 관련 언어들에 어떠한 영향을 보여주는 것으로 그들의 임무를 모두 마쳤습니다. 긴 시간동안 실행되어도 아무 문제 없고, 모든 종류의 컴퓨터와 출력 장치와도 잘 어울릴 수 있는 새로운 버전의 T_EX과 METAFONT를 작성해야 한다는 것입니다. 게다가 새 버전 프로그램들의 소스 코드는 책 형태로 출간되어야 합니다. 왜냐하면 T_EX은 이미 프로그램 소스 코드도 최고의 품질로 문서화 할 수 있기 때문입니다. 나는 가능한한 매우 이식성이 높은 프로그램을 작성하고자 최선을 다했으며, 아울러 누구나 읽기 쉽도록 프로그램하고자 했습니다. 그렇게 함으로 해서, 그 프로그램들이 중요하고 쓸모있다고 판단되는 사람들이 그 프로그램을 이어 받아서 개선하거나 유지 보수 할 수 있도록 하기 위함입니다. 이렇게 함으로 해서 만약 나의 조판에 관한 연구들이 누군가에게 끌리는 점들이 있다면, 그들은 나의 연구를 이어받아 그들 필요에 맞는 일을 수행할 수 있을 것이라는 것을 알기 때문에, 나는 떳떳하게 T_EX이외의 다른 하고자 하는 일을 아무런 거림낌 없이 당당하게 할 수 있었다고 생각했기 때문입니다.

그래서 그것이 나의 새로운 목표가 되었습니다. 나는 이 일을 이루기 위해서는 앞으로 1년 혹은 2년의 기간이 더 필요하다고 생각하게 되었습니다. 오리지널 T_EX 프로그램은 T_EX78 이라고 이름붙였고, 새로운 버전은 T_EX82 라고 부르게 되었습니다.

1981년 상반기는 강의와 그 밖의 많은 잡무로 때문에 너무 바빠서 다른 일을 도저히 할 수가 없었습니다. 그래서 8월 22일 부터 T_EX82를 작성하기 시작했습니다. 9월 9일, 나는 DOC 프로그램을 완전히 다시 작성해야 할 필요를 느꼈서, 그 프로그램을 새롭게 개선하여, WEB 이라는 새로운 프로그램을 만들어 내는데 두 달가량을 소비했습니다. 그 이후로 WEB은 내가 프로그래밍 할 때마다 사용하는 가장 좋아하는 프로그래밍 언어가 되었습니다. (WEB은 DOC과 달리, 자기 자신으로 작성된 독특한 놈입니다.) 유럽에서 한 달가량을 보낸 후인 1982년 12월 1일에 T_EX82를 작성을 다시 시작했습니다. T_EX82의

초안은 1982년 6월 29일에 완성되었고, 예전처럼, 그 프로그램을 모두 완성하기 전까지는 테스트를 위해서 작성 중인 프로그램을 실행하지 않았습니다.

그러는 동안, 나에게는 다른 걱정 거리가 생겼습니다. 1981년 1월, *Seminumerical Algorithms*이 드디어 책으로 만들어져서 그것을 받아봤는데, 그때 나는 내심 그러한 어려운 주제에 관한 책일지라도 폰트가 적절히 배치되고 책의 구성이 아름다운 책으로 되었으려니 했는데, 그 책은 나의 기대를 완전히 저버려서 나는 절망감에 휩싸였습니다. 그래서 나는 폰트에 대해서 배워야 할 것들이 너무도 많다는 것을 깨달았습니다. 처음 만들어낸 Computer Modern type들을 책으로 봤을 때의 느낌은 “이건 아니잖아!” 였습니다. 그 폰트들은 저해상도에서는 그럭저럭 봐줄만해서 높은 해상도에서도 그러려니 했는데, 전혀 그렇지 않았습니다. 나는 폰트에 관해서는 아무것도 모르는 걸음마 단계였던 것이었습니다. 그래서 1981년 말에 나는 내게 부족한 전문 지식을 채워줄 폰트에 관한 전문가 Richard Southhall을 만나서 스탠포드로 초대했습니다. 우리는 1982년 4월 한달동안을 하루에 16시간씩 일을 하면서 보냈는데, 그때 Computer Modern 을 A 부터 z까지 완전히 다시 만들었습니다.

82년 여름부터 T_EX82를 디버깅했고, 10월에 지금은 *The T_EXbook* 이라고 알려진 T_EX 매뉴얼을 작성하기 시작했습니다. 앞에서 언급한 처음 매뉴얼은 단지 21일 만에 작성된 것이었습니다. 하지만 이번의 *The T_EXbook*은 보다 높은 수준의 요구를 충족시키고자 했기에 1년을 완전히 다 바쳐서야 매뉴얼 작성을 마칠 수 있었습니다.

T_EX이 드디어 완숙한 프로그램이 된 시기는 82년 10월 과 83년 10월의 1년 동안이었습니다. 나는 그 매뉴얼을 작성면서 머리 속으로는 계속 T_EX의 디자인의 모든 면면을 다시 생각해 보았습니다. 이 과정에서 다행히도, 매우 우수한 지원자들로 구성된 모임의 도움으로 받을 수 있었습니다. 우리는 매주 금요일 정오에 두 세시간씩 만남을 가져서 T_EX의 중요한 기능에 있어서 트레이드오프에 관해서 논의했습니다. 다양한 배경 지식을 가지고 있는 그 모임의 사람들은 매우 중요한 조언을 해주었습니다. 마침내 1983년 2월 9일, 나는 나의 endgame 전략의 첫번째 단계가 성공적으로 마무리 되었다는 것을 알아서, 감사하는 마음으로 36명의 손님들을 초대하여 Palo Alto의 Fuki-sushi 식당에서 T_EX 파티를 열었습니다.

이제부터는 조판 관련 프로그램의 역사에 대해서 말씀드리겠습니다. 나는 1983년 12월과 1984년 1월에 걸쳐서 WEB으로 METAFONT를 작성했고, 1984년 8월부터 1985년 10월에 METAFONT*book*을 집필했습니다. 그 기간의 다섯

달은 METAFONT로 Computer Modern을 다시 작성하는데 소비했습니다. 나는 T_EX 프로젝트가 모두 끝난 1985년 10월에 또다시 안식년을 맞이하게 되었습니다. 몇가지 최종작업을 마친 후에 드디어 1986년 5월 21일 그동안 길고 길었던 나의 “endgame”의 종료를 선언하는 기쁨을 가질 수 있었습니다. 그때 나의 출판사가 후원하여 보스턴에 있는 Computer Museum에서 리셉션을 열었는데, 이때 나는 *Computers & Typesetting* 이라는 다섯 권으로 된 하드 커버의 책을 처음 보았습니다. 그 책들은 T_EX, METAFONT, Computer Modern에 관한 9년 동안의 나의 연구 결과를 총망라한 책들입니다.

그 후 1년여의 시간을 더 보낸 다음, 나는 T_EX은 100% 완벽한 프로그램이라고 하고 싶었습니다. 그러나 그렇게 할 수 없었고, 그건 지금까지도 마찬가지입니다. 나는 지난 1월에 그동안 숨겨져 있던 T_EX의 오동작에 휘청거리야 했습니다. 그리고 T_EX의 내부 구조를 기반으로 해서 대형 소프트웨어의 개발에 대해서 강의를 했는데, 그 강의의 학생들이 T_EX의 여러가지 개선사항을 말해주기도 했습니다. 그래서 나는 T_EX에는 아직도 발견되지 않은 버그가 잠복해 있다고 인정해야만 했습니다. 그 당시 나는 내가 지금 쓰고 있는 이 논문 작성을 1, 2년 동안 멈추기로 했습니다. 그래서 나는 현재 나의 에러나 버그 목록을 기록해 둔 로그책이 이제는 믿을 만하다고 느끼게 되었습니다.