

컴퓨터를 과학의 경지로

컴퓨터를 과학의 경지로
— 도널드 크누스

로버트 슬레이터 지음 · 이규창 옮김

2006

동아일보사 · 한국컴퓨터전문학교

일러두기

- 이 글은 『컴퓨터 영웅들』 가운데 마지막 쪽지를 그대로 옮긴 책입니다. 이 책에서는 찰스 배비지를 필두로 콘라드 쥬제, 데니스 리치와 케네스 톰슨, 빌 게이츠, 스티브 잡스를 비롯하여 마지막으로 도널드 크누스에 이르기까지 34인의 컴퓨터 영웅들을 소개하고 있습니다.
- 원저는 로버트 슬레이터(Robert Slater)씨가 1987년에 쓴 PORTRAITS IN SILICON이고, 한국어판은 이규창씨가 옮겨 1993년 4월 초판이, 1995년 3판이 동아일보사에서 나왔습니다.
- KTUG FAQ wiki의 [Donald Knuth](#) 페이지에 조성민님이 소개한 Knuth에 관한 다른 글이 있으니 참고하시기 바랍니다.
- 원문에는 없는 설명과 그림을 Progress가 아주 죄금 덧붙였습니다.

Copyright © 2006, 이규창.

도은이아빠님의 KTUG Collection 2006 테스트 버전을 이용하여 조판함. 사용한 T_EX 환경은 W32T_EX/Ko이고, 홍석호님의 NTEmacs를 편집기로 사용함.

KTUG, Korean T_EX Users Group

도널드 크누스의 이름을 거론할 때면 으레 다음과 같은 수식어가 뒤따른다. ‘세계 컴퓨터과학계의 거장’ 이름까지 딸 것도 없이 크누스라는 성만으로도 그의 탁월함은 타인을 압도한다. 크누스(‘크’ 발음을 한다)는 분명 비좁은 연구실에서 북적대며 열심히 연구하는 전형적인 대학교수이지만 그가 뛰어난 작가요, 작곡가였다 인쇄술의 전문가라는 사실에 새삼 놀라지 않을 수 없게 된다. 첫 논문이 『매드』(mad)라는 잡지를 통해서 발표된 것부터 엿새 만에 단편 소설을 써냈다는 이야기에 이르기까지 그를 둘러싼 이미지는 대부분 평범을 훨씬 넘는 것이다. 무엇보다도 그는 컴퓨터과학의 바이블이라 칭송되는 『컴퓨터 프로그래밍 기술』(The Art of Computer Programming) 시리즈로 유명하다. 한 평론가는 이 시리즈가 기하학에서 유클리드의 저술만큼이나 컴퓨터과학에서 중요한 자리를 차지하는 작품이라고 주장한다.

1980년대 중반까지 크누스는 계획된 7권의 책 중에서 1968년, 1969년, 1973년 세권만을 출간하는 데 그쳐 당초 장대한 계획의 절반 정도밖에 도달하지 못했다. 그럼에도 불구하고 그는 출판사들의 원성을 들으면서까지 저술활동을 뒤로 미룬 채 식자(植字)의 컴퓨터화라는 전인미답의 영역에 뛰어들어 9년의 세월을 보냈다. 이 기간 동안의 그의 작업은 인쇄술 분야의 혁명을 가져왔다.

도널드 크누스는 1938년 1월 10일 위시콘신주 밀워키에서 태어났다. 아버지는 고등학교 학생들에게 부기를 가르쳤으며 집 지하실에 인쇄소를 차려놓고 있었다. 어릴 적 크누스는 열자리 숫자끼리의 곱셈이 가능하지만 답이 나오려면 20초 가량 걸리는 아버지의 레밍턴랜드 계산기를 즐겨 갖고 놀았다. 그는 시행착오를 겪어가면서 10의 평방근을 얻으려 고심했던 기억을 지금도 갖고 있다. 그 답이 3.16과 3.17 사이에 있다는 것을 발견하고는 이 결과가 파이(π)의 진정한 값

2 컴퓨터를 과학의 경지로

이며 교과서의 3.14는 틀렸다고 굳게 믿었다. 그러나 그는 곧 자신의 결론이 틀렸다는 것을 알게 되었다.

숫자와 마찬가지로 단어도 그를 사로잡았다. 7~8학년 동안 그는 문장을 도해하면서 대부분의 시간을 보냈다. 영어산문의 문장들은 쉽게 이루어졌지만 찬송가는 도해가 거의 불가능해 보였다. 밀워키의 한 사탕회사가 ‘제이글러의 큰 사탕’(Zeigler’s Giant Bar)이라는 문구에 나오는 문자로 얼마나 많은 단어를 만들 수 있는가를 질문하는 컨테스트를 주최한 적이 있었다. 그는 병을 빙자해 2주 동안 집에 틀어박힌 채 대사전을 뒤져가며 4500개까지 찾아냈다. 이것은 심사위원들이 뽑은 2500여개보다 훨씬 많은 것이었다. 나중에 그는 아포스토프(‘)를 사용하면 더 많은 단어를 만들 수 있다는 것을 알았다. 상품으로 그에 주는 썰매가, 그의 학급에는 TV가, 그리고 같은 반 친구 모두에게는 바로 그 사탕이 수여됐다.

수학에 대한 크누스의 관심은 고교 1학년 시절 무한한 가능성의 영역인 대수 방정식의 그래프에 매료되면서 높아지기 시작했다. 크누스는 누구든지 그에게 연관된 직선을 몇 개 주면 곧 이에 대한 방정식의 근을 구해줄 수 있는 시스템을 개발했다. 이렇듯 어렸을 적 그래프에 대한 관심이 후에 컴퓨터를 이용한 인쇄시스템 ‘메타폰트’ 개발의 기초가 되었다.

물리학에 관심을 갖게 되면서 크누스는 그전에 좋아했던 음악과는 거리가 멀어졌다. 그는 피아노를 굉장히 좋아했고 몇편의 오케스트라를 작곡한 적도 있었다. 약간의 열등 콤플렉스를 갖고 있었던 크누스는 바로 그것이 원동력이 되어 열심히 공부했다. 밀워키 루터고등학교 3학년 당시 그는 수학 점수가 나빠 대학에 못 갈지도 모른다고 고민했다. 그가 평균 97.5점을 받아 최고 성적으로 졸업을 했다는 사실로

미루어볼 때 믿기 어려운 사실이다.

1956년 그는 오하이오주 클리블랜드의 CIT 물리학과에 입학했다. 고등수학에 처음 접한 것이 1학년 미적분학 시간 때였다. 또다시 할 수 없을 것이라는 불안감 때문에 크누스는 미적분학과 분석기하학¹에 별도의 시간을 할애해서 열심히 공부했다. 1학년 때인 1957년 6월에 『매드』라는 잡지에 그는 자신의 첫 논설을 기고했다. 제목은 “무계와 거리측정단위의 포트르저비 시스템”이었다. 이 매드식(式)의 단위는 기존의 미터법에 대한 대안을 제시한 것인데, 예를 들면 100만 포트르저비(potrzenie)는 퍼슐러기너(furshlugginer) 포트르저비이고 100만번째 포트르저비는 파시멜트(farshmmelt) 포트르저비이다. 이 시스템은 아주 완벽하게 갖추어져 있어서 카우즈놉스키(cowznowski) 당 100클라스(日)를 기초로한 달력까지 만들어놓았다.

대학 2학년에 올라가기 전 여름방학에 크누스는 처음으로 컴퓨터를 접했다. 그는 방학 동안 연구소에서 통계기사로 그래프를 작성하는 일을 맡았다. 당시 그 옆방에는 IBM 650이 한대 설치되어 있었다. 그는 이 기계에 흠뻑 빠져들어 훗날 자신의 대표적인 저서인 『컴퓨터 프로그래밍 기술』을 이 연구소의 IBM 650 컴퓨터에 바친다고 적어넣을 정도였다. 심지어 일부 선생님들은 그에게 컴퓨터에 너무 집착하지 말라고 충고까지 할 정도였다. 그러나 그는 이미 호기심이 발동한 상태였다. 누군가가 그에게 IBM 650의 작동법을 설명해주자 그 컴퓨터의 반짝이는 불빛에 매료되어 며칠이고 컴퓨터에 매달려 밥냏없이 마라톤 학습을 계속하곤 했다.

1957년 가을에 그는 이론 수학에 특히 심취했다. 한번은 교수가 학생들에게 해답이 가능한지는 밝히지 않고 특별한 문제를 하나 숙제

¹ 아마도 ‘해석기하학’(analytic geometry)를 이렇게 번역한 듯

로 내주었다. 문제를 풀든 안 풀든 자유였지만 푸는 사람은 무조건 A학점을 받게 되어 있었다. 어느 누구도 이 불가능해 보이는 도전에 무모하게 덤벼려하지 않았다. 그러던 어느 토요일 밴드부의 일원이었던 크누스는 행사장으로 가는 버스를 놓치고 남은 시간을 보내기 위해 그 문제를 풀어보기로 결심했다. 그는 문제의 답을 다음 주에 제출하고는 그 과목에 A학점을 받았다. 그러나 크누스는 이유 없는 죄책감 때문에 남은 학기 동안 수업을 듣지 않았다. 몇년이 지난 후 그 교수에게 문제에 대한 새로운 해답을 제출했다.

한편 크누스는 쉽게 물리학을 포기했다. 그의 실험 실습능력은 별로 신통치 않아서 기기들을 바닥에 떨어뜨리거나 가장 늦게 실험을 마치는 일이 종종 있었다. 193cm의 키에 용접용 책상이 터무니없이 낮았던 그는 안경 밑으로 초점을 맞추기가 참으로 어려웠다. 또는 전기충격에 굉장히 민감해서 깜짝깜짝 놀라곤 했다. 결국 2학년 말에 물리학을 포기하고 수학으로 전공과목을 바꿨다.

크누스는 자신이 간사로 있던 대학 농구팀을 위해 컴퓨터 실력을 발휘했다. 그는 선수 개개인을 평가하는 공식을 만들어서 그 선수의 총득점보다는 팀에 대한 공헌도를 점수화하는 작업을 하였다. 크누스는 IBM 650 앞에 앉아서 통계수치를 입력하고 코치는 옆에서 이 모습을 지켜보고 있었다. 그 프로그램을 이용하여 코치는 가장 우수한 선수를 골라낼 수 있었고 다음 경기에는 그들을 기용했다. 이것은 선수들로 하여금 분발하게 하는 효과도 가져왔다. 1960년에 그의 팀은 리그 우승의 영광을 안았다. 크누스의 이 '마법의 공식'은 뜻하지 않게 일반의 관심을 끌어 월터 크롱카이트의 일요일 뉴스 프로그램과 『뉴스위크』지에 나가기도 했다.

크누스는 1960년 최우등으로 수학 학사학위를 받았다. 동시에 아

주 인상적인 학생으로 선정되어 교수부로부터 석사학위까지 동시에 취득하는 유례없는 특전을 받기도 했다. 1960년 여름 그는 컴퓨터업계에 새로 참여한 업무용 기기업체 버로즈사(社)의 패서디나 사무실에서 소프트웨어를 작성하고 하드웨어를 분석하는 일을 했다. 알골 컴파일러 하나를 만들어주고 그는 5500달러를 받았는데 나중에 그는 자신이 얼마나 부당하게 대우받았는지 알게 되었다. 사실 소프트웨어 회사들은 다른 프로그래머들에게는 그 액수의 몇배를 지불하고 있었던 것이다.

1960년 가을 그는 캘리포니아 공과대학의 박사과정에 입학했다. 이 대학원생이 컴파일러를 만들 줄 안다는 소문이 들리자 1962년 1월 애디슨-웨슬리 출판사는 크누스에게 그런 주제로 책을 써줄 것을 의뢰했다. 그 제안이 마음에 들었던 그는 12개 장의 제목을 대충 정해 보았다. 그중 처음 3개장은 그해 가을 강의를 위해 준비한 노트였다.

1963년 6월 크누스는 수학 박사학위를 받았다. 그리고 대학의 부교수로 임용되었다. 그 주제에 대해서는 실상 아는 것이 별로 없었지만 곧바로 그는 책을 쓰기 위한 작업에 착수했다. 아직 초창기의 컴퓨터과학은 허점투성이 책들 때문에 어려움이 많았다. (크누스가 즐겨 말하는 것처럼 컴퓨터과학은 0과 1의 패턴에 대한 연구에 불과한 것이다.) 이 분야에 대한 연구를 시작할 때 그는 대다수 기존의 문헌들이 틀리게 쓰여져 있다는 것을 알았다. 수많은 프로그래머들이 끊임없이 새로운 연산을 개발해내고 있었지만 좋은 아이디어가 나와도 종종 판매 부수가 적은 잡지나 공학논문 속에 묻혀 사라지곤 했다. 수많은 프로그램들이 읽히지 않은 채 지나쳐버렸다. 이미 발견해놓고도 이렇게 과묵히 주목을 받지 못한 공학과 방법론들을 다시 발굴해야 한다는 게 그의 주장의 요점이었다. 그는 가치 있는 문헌들 중 일부

를 요약하면 쓸모가 있을 것이라는 생각이 떠올랐다. 이전의 컴퓨터 프로그래밍을 요약하고자 했던 사람들은 그들 자신의 이론에 집착한 나머지 일반성을 잃고 있었다. 새로운 아이디어를 개발하지 않고서도 단지 글재주만으로 큰 일을 해낼 수 있다고 생각한 크누스는 자신이 그 작업의 적격자임을 확신했다.

이 기념비적인 작업의 목적은 컴퓨터 프로그램을 개발한 사람들을 위해 그들의 대변자가 되고 이 분야에 현존하는 지식을 견고한 틀로 구축하는 데 있었다. 그의 주요한 업적은 자료를 체계적으로 분류하고 방법론을 분석하는 데 있었다. 그는 여러가지 방법론 중에 최적의 이론들을 개발하고 이 이론들간의 간격을 좁히고자 노력했다. 그는 이 간격을 처음으로 인식해 비조직화된 자료들을 통합하려 했던 최초의 인물이었다. 다행히에 그 간격을 메우는 데는 그리 큰 어려움이 없었기 때문에 그는 이 작업을 처음으로 달성한다는 기쁨에 사로잡혔다. 만일 해결할 수 없는 문제가 발견되면 연구해야할 과제로 서술해 놓으면 되었다.

처음에 그는 컴파일러에 대한 책을 한권 쓰겠다고 생각했었다. 그러나 몇개의 장에 대한 초안을 잡으면서 애초의 생각보다 범위가 훨씬 더 커지리라는 것을 알게 됐다. 출판사의 허락을 얻어 그는 쓰고 또 썼다. 1965년 6월 마침내 12개 장으로 이루어진 초안을 완성했다. 이는 무려 3000페이지 분량의 방대한 것이었다. 같은해 10월 그는 애디슨-웨슬리사에 제1장을 보냈다. 출판사의 편집장은 이 장의 길이로 추정할 때 이 책은 약 2천페이지 정도일 것이라고 판단했다. 출판사는 그에게 12개의 장은 총 7권으로 나누어지며 권당 1개 내지 2개 장으로 구성될 것이라고 설명했다.

크누스는 이 책을 쓰면서 약간 장난을 쳤다. 1권의 색인 중 순

환논리라는 항목을 찾은 독자들은 “‘논리, 순환’을 보시오”라고 쓰여 있는 것을 보고 ‘논리, 순환’ 항목으로 가면 다시 “‘순환, 논리’를 보시오”라고 돼 있는 것을 발견한다. 그런데 사실 이것은 컴퓨터의 가장 큰 맹점 중의 하나인 무한고리를 설명하는 크누스의 계산된 방식이었다. 제1장은 제1권의 절반 정도를 차지했고 제2권은 임의수(random number)에 관한 제2장과 원래 ‘잡다한 유틸리티 루틴’이라는 제목의 제6장이 합쳐져 이루어졌다. 제2권을 쓰는 데 밤낮을 가리지 않고 매달렸던 그는 1967년 여름 중증궤양에 시달리기까지 했다. 그는 당시의 노고를 아직도 분명히 기억하고 있다. 이제 제2권의 333페이지까지 완성하여 유클리드의 연산 중간까지 와 있는 상태였다.

이 시리즈는 에디슨-웨슬리사의 베스트셀러 중 하나로 1970년대 중반부터 1980년대 중반까지 3권의 책이 각각 매달 2000권씩 팔렸다. 이 책은 중국어, 루마니아어, 일본어, 스페인어 그리고 러시아어로 번역되었고 폴란드어와 헝가리어판이 계획중에 있다.² 컴퓨터기기협회는 1974년 그에게 튜링상을 수여하면서 “그의 저서들은 옛날부터 품어왔던 일련의 발상들을 현대 컴퓨터과학자들의 표준 관행으로 전환시키는 데 크게 공헌했다. 이 저서들에 나타난 기법, 연산, 관련 법칙의 총합은 컴퓨터과학을 조직화하는 힘이 됐고 또 교과과정을 개발하는 중심들의 역할을 수행해왔다”고 축사를 바쳤다. 그에게 영광이 계속되는 가운데 1979년에는 카터 대통령으로부터 국가과학훈장을 받기도 했다.

크누스는 큰 몸집에 아주 침착한 성격을 가진 인물이지만 언변이 유창했고 손놀림도 재빨랐다. 그의 주된 관심사 중의 하나는 음악이었다. 그는 캘리포니아주 멘로 파크의 베서니 루터파 교회에 있는

²류 광이란 분께서 한국어판을 번역중인 것으로 압니다. [TeXbook 번역 홈페이지](#)

- Car: LISP terminology for the first component of a List; analogous to INFO and DLINK on page 411, or to ALINK on page 415.
- Cards, playing, 51, 69, 233-237, 238, 377-378.
- Cards, punched, 136-137, 152, 229.
- Carlitz, Leonard, 96, 499, 506.
- Carlyle, Thomas, xiv.
- Carpenter, Brian Edward, 229.
- Carr, John Weber, III, 458.
- Case Institute of Technology, i.
- Cassini, Gian (= Giovanni = Jean) Domenico (= Dominique), 81.
- Catalan, Eugène Charles, 407.
- numbers, 407.
- Cate, Esko George, 523.
- Cauchy, Augustin Louis, 92, 475, 490, 506, 520, 582.
- inequality: $(\sum a_k b_k)^2 \leq (\sum a_k^2)(\sum b_k^2)$, 36.
- matrix, 37-38, 475.
- Cayley, Arthur, 396, 406-407, 586, 597.
- CDC 1604 computer, 124, 529.
- Cdr: LISP terminology for the remainder of a List with its first component deleted; analogous to RLINK on page 411, or to BLINK on page 415.
- Ceiling function $\lceil x \rceil$, 39, 41.
- Cell: A word of the computer memory, 127.
- Cellar, 240.
- Central moment of a probability distribution, 105.
- Centroid of a free tree, 387-388, 397, 589.
- Chain: A word used by some authors to denote a linked linear list, by others to denote a linearly ordered set.
- Chain rule for differentiation, 52.
- Chaining: A word used by some authors in place of "linking".
- Chakravarti, Gurugovinda (उदकगविकिद चक्रवर्ती), 53.
- Chen, Tien Chi (陳天機), 471.
- Cheney, Christopher John, 421.
- Cheney, Ednah Dow Littlehale, 377.
- Chernoff, Herman, 502.
- Chess, 6, 194, 272.
- Chia Hsien (賈堯), 53.
- Child link, 427-433.
- Children in tree structures, 311, 317, 334-335, 352.
- Chowla, Paromita (পারমিতা চাওলা), 307.
- Christian IX, King of Denmark, 310, 311, 562.
- Christie Mallowan, Agatha Mary Clarissa Miller, xvii.
- Chu Shih-Chieh (= Zhū Shijié, Zhū Sòngtíng; 朱世傑, 朱松庭), 53, 59, 70.
- Chung Graham, Fan Rong King (鍾金芳蓉), 514.
- Chung, Kai Lai (鍾開榮), 105.
- CI: The comparison indicator of MIX, 126, 134, 142, 211, 213, 228.
- Circle of buffers, 218-227, 231.
- Circuit, Eulerian, in a directed graph, 374-376, 379-380, 584.
- Circuit, Hamiltonian, in a directed graph, 374, 379.
- Circular definition, 263, 308, *see* Definition, circular.
- Circular linking, 273-280, 302, 355, 411, 459.
- Circular store, 240.
- Circulating shift, 135.
- CITRUS subroutine library, 457.
- Clark, Douglas Wells, 604.
- Clavius, Christopher, 159.
- Clock, for real time, 228.
- Clock, simulated, 283, 288.
- Clock, solitaire game, 377-378.
- Closed subroutine, *see* Subroutine.
- CMath: *Concrete Mathematics*, a book by R. L. Graham, D. E. Knuth, and O. Patashnik, 11.
- CMP1 (compare r11), 134, 210-211.

그림 1: *see* 'Circular definition'. 그림 2를 보세요. (Amazon에서 발췌한 *The Art of Computer Programming* vol. 1의 index 부분)

약 1000개의 파이프로 이루어진 바로크풍의 파이프 오르간을 직접 설계했고 자기 집에 이것의 축소판을 설치해놓기도 했다. 1968년부터 그는 스탠퍼드 대학에 컴퓨터과학 교수로 재직하였다.

크누스는 또한 소설가이기도 하다. 그의 소설 『초현실적 숫자들:

- Representation in a precise, formalized language of some facts or concepts, often numeric or alphabetic values, to facilitate manipulation by a computational method, 215.
packed, 128, 158.
- Data organization: A way to represent information in a data structure, together with algorithms that access and/or modify the structure.
- Data structure: A table of data including structural relationships, 232–465.
linear list structures, 238–298.
List structures, 408–423.
multilinked structures, 424–434.
orthogonal lists, 298–307, 424–434.
tree structures, 308–408.
- Daughter, in a tree structure, 311.
- David, Florence Nightingale, 66.
- Davies, David Julian Meredith, 445.
- Davis, Philip Jacob, 50.
- Dawson, Reed, 584.
- de Bruijn, Nicolaas Govert, 121, 122, 375, 379, 380, 478, 504, 543, 565.
- de Moivre, Abraham, 74, 83, 87, 106, 182, 536.
- De Morgan, Augustus, 17.
- Deallocation, *see* Liberation.
- Debugging: Detecting and removing bugs (errors), 192–193, 201, 257, 297, 413, 556.
- DECI (decrease r1), 134, 210.
- DECA (decrease rA), 134, 210.
- Decimal computer: A computer that manipulates numbers primarily in the decimal (radix ten) number system.
- Decimal number system, 21, 619.
- DEX (decrease rX), 134, 210.
- Defined symbol, an assembly language, 153.
- Definition, circular, *see* Circular definition.
- Degree, of node in tree, 308, 317, 377.
of vertex in directed graph, 372.
- Deletion of a node: Removing it from a data structure and possibly returning it to available storage.
from available space list, *see* Reservation.
from deque, 251, 297.
from doubly linked list, 281, 290–291, 297.
from doubly linked ring structure, 358.
from linear list, 239.
from linked list, 236, 255, 276, 305.
from queue, 242, 244–245, 254, 261, 265, 273–274.
from stack, 241, 242, 244–245, 247, 254, 259, 269, 273–274, 278, 458.
from tree, 358.
from two-dimensional list, 305.
- Demuth, Howard B., 120.
- Depth-first search, 578, 581.
- Deque: Double-ended queue, 239–243, 269.
deletion from, 251, 297.
input-restricted, 239–243, 416.
insertion into, 251, 297.
linked allocation, 280, 297.
output-restricted, 239–243, 269, 274.
sequential allocation, 251.
- Derangements, 180, 183.
- Derivative, 90, 338.
- Dershowitz, Nachum, 518, 588.
- Descendant, in a tree structure, 311.
- Determinant of a square matrix, 37–39, 81, 378–379, 382.
- Deuel, Phillip DeVere, Jr., 556.
- Deutsch, Laurence Peter, 418, 421, 422.
- Dewar, Robert Berriedale Keith, 614.
- Dewey, Melvil, notation for binary trees (due to Galton), 317, 331, 346, 405.
notation for trees, 313, 317, 382–383, 460.
- Diaconis, Persi Warren, 491.
- Diagonals of polygons, 408.
- Diagrams of structural information, 234, 279.
before-and-after, 260–261, 278, 281, 571.
binary trees, 312, 318, 563.
List structures, 315–317, 408–409.
tree structures, 309–315, 337, 346, 349, 460.
- Dickman, Karl Daniel, 525.
- Dickson, Leonard Eugene, 81.
- Dictionaries of English, 1–2, 215–216.
- Differences of polynomials, 64.
- Differentiation, 90, 338–347, 459.
chain rule for, 52.
- Digamma function $\psi(z)$, 44, 75, 493.
- Digit: One of the symbols used in radix notation; usually a decimal digit, one of the symbols 0, 1, ..., or 9.
- Digraph, 372, *see* Directed graph.
- Dijkstra, Edsger Wijbe, 17, 191, 230, 231, 240, 459, 462, 545, 580, 605.
- d'Imperio, Mary Evelyn, 462.
- Directed graphs, 372–374, 422.
as flow charts, 364–365, 377.
balanced, 374–375.
connected, 363.
regular, 379.
strongly connected, 372, 377.
- Discrete system simulation, 203, 282–298.
synchronous, 282, 298.
- Disjoint sets: Sets with no common elements.
- Disk files, 136–137, 435, 463.
- Disk input, buffered, 228.
- Disposal, *see* Garbage collection, Liberation.
- Dissection of a polygon, 408.
- Distributed-fit method of storage allocation, 450–451, 456.

그림 2: *see* ‘Definition, circular’. 그림 1을 보세요.

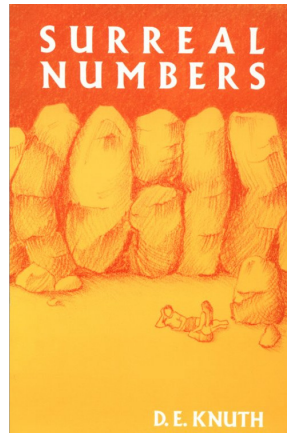


그림 3: 『초현실적 숫자들』, Amazon에서 발췌

어떻게 두 명의 퇴학생이 이론 수학에 눈을 돌려 진정한 기쁨을 발견했는가?³는 1974년 에디슨-웨슬리사를 통해 출판되었다. 이 책은 케임브리지 대학의 존 호턴 컨웨이에 의해 창안된 새로운 수 체계의 탐험을 주요 내용으로 하고 있다. 크누스는 1972년 어느날 컨웨이와 함께 점심을 하면서 이 체계에 대한 지식을 얻었다고 한다. 그리고 나서 잠을 자다가 갑자기 벌떡 일어나서는 아내에게 이 소설이 떠올라 곧 써야 되겠다고 이야기했다. 그러나 그녀는 걱정하지 않아도 됐다. 왜냐하면 컴퓨터 프로그래밍 시리즈와 달리 이것은 1주일만에 끝났기 때문이다. 1972년 12월 안식년 휴가 때 그는 입센이 희곡을 쓰던 노르웨이의 오슬로 시내 한 호텔에 방을 잡아놓고 소설을 완성했다. 이

³*Surreal Numbers: How two ex-students turned on to pure mathematics and found total happiness*, Illustrated by Jill C. Knuth

소설의 두 주인공은 쾨웨이의 수 체계를 탐험하고 완성해가는데, 이는 작품을 쓰면서 크누스가 직접 경험한 것이기도 했다. 어떤 비평가는 이 소설이야말로 중요한 수학적 발견이라고 지적했다. 크누스 자신은 쾨웨이의 이론을 가라츨기 위한 것이 아니라 어떻게 그러한 이론이 만들어지는가를 설명하기 위해서 쓴 것이라고 밝혔다.

1977년 봄에 크누스는 아무런 사전 통보도 없이 중도에 직업을 바꿨다. 그는 컴퓨터 프로그래밍 시리즈 제2권의 수정판 교정쇄를 지켜 보면서 갑자기 책 인쇄술에 일대 혁신이 필요함을 느꼈다. 그는 아주 형편없이 보이는 교정쇄들을 없애려고 했다. 그것들은 공간 배열이 조악했고 특히 표준문자와 수학 방정식이 함께 나오는 책일 경우에는 문제가 심각했다. 크누스는 활자의 광학 전사를 사용하는 인쇄직이 왜 그렇게 인기가 없는지 알고 싶었다. 결국 그는 컴퓨터과학과 수학을 적용해 서정의 모양을 개량하는 일에 몇개월을 투자하기로 결정했다. 그러나 이 일에는 장장 9년의 세월이 소요되었다.

크누스는 최초의 컴퓨터 인쇄용 표준언어인 $\text{T}_{\text{E}}\text{X}$ 과⁴ 수학을 사용하여 알파벳을 만들어내는 메타폰트 시스템을 개발해냈다. $\text{T}_{\text{E}}\text{X}$ 은 식자 역사에서 가장 중요한 발명의 하나로 간주되었다. 일부에서는 이를 구텐베르크의 금속활자에 견줄 만한 것으로 평가하기도 했다.

$\text{T}_{\text{E}}\text{X}$ 는 식자공들이 문자와 심벌을 배치할 때 이제껏 불가능했던 유용성과 미적 특성을 발휘할 수 있게 했다. 크누스 체계에서 원판은 컴퓨터 스크린에 검고 흰 점의 격자로부터 전자적으로 만들어진다. 처음에 컴퓨터는 디지털 언어를 사용해서 어디에 점을 위치시킬 것인가를 지시받는다. 그후 광학적으로 복사가 이루어지면 인쇄를 위한 모판이 만들어진다.

⁴원문에는 ' $\text{T}_{\text{E}}\text{X}$ 와'로 되어 있었습니다.

메타폰트는 조작자에게 폰트를 전자적으로 만들어낼 수 있게 했다. 폰트는 비디오 터미널에 입력되어 곧바로 여러 방식으로 변화될 수 있다. 과거 식자공들은 하나의 폰트를 만드는 데 몇개월에서 몇년까지 걸리곤 했다. 그들은 각 문자마다 각각 수십개의 활자서랍을 갖고 있어야 했다. 하나의 문자에 약간의 변화가 있어도 이것과 형태, 크기, 배열에서 상호 관련되는 여타의 문자 모양에도 영향을 미치게 되기 때문이었다. 그러나 메타폰트를 이용한 후에는 한번의 명령으로 똑같은 결과를 얻을 수 있게 되었다.

크누스는 이 두 프로그램을 공공분야에 제공했다. 그와 스탠퍼드 대학 둘 다 이들로부터 한푼도 받지 않았다. 그는 단지 책에 대한 사랑으로 이 작업을 했고 필요한 곳에 제공했을 뿐이라고 말했다. “나는 이미 명성을 얻었다. 내가 쓴 책은 잘 팔리고있다. 나는 내가 좋아서 한 일에 소유권을 주장하고 싶지 않다. 게다가 수학자들은 자기의 증명에 대한 대가로 돈을 받는 데 익숙하지 않다.”

크누스가 메타폰트와 T_EX을 사용하여 『컴퓨터 프로그래밍 기술』 제2권을 재구성했을 때 그 결과는 훨씬 나아지기는 했으나 완벽한 것은 아니었다. 숫자의 모양이 약간 일그러진 상태였다. 그래서 그가 일류 그래픽 디자이너들과 이 새로운 프로그램을 개량하고 완벽하게 꾸미는 데 또 5년의 세월이 흘렀다. 1986년 봄 인쇄술에 대한 그의 연구는 끝이 나고 5권으로 된 『컴퓨터와 식자술』⁵이 출간되었다. 제1권은 T_EX용 사용자 지침서이고 제2권은 T_EX용 코드 목록집이며 제3권과 제4권은 메타폰트용 사용자 지침서와 코드 목록집이고 제5권은 메타폰트 프로그래밍의 500가지 예문을 기술해놓고 있다.

⁵ *Computers & Typesetting Series. The T_EXbook* 페이지 참고. 2006년 작은 나무님의 **택북 읽기** 페이지는 절찬 연재중이다.

크누스는 컴퓨터과학을 독립 분야로 만들기 위해 누구보다도 많은 일을 했다. 그는 “컴퓨터 과학자들이 서로 몽치려면 서로 대화할 수 있는 학술적 장이 마련되어야 한다. 이제 우리는 이름을 갖게 됐다. 과거 우리들 가운데 일부는 물리학자로 또는 수학자로 불렸으나 그 어느 분야에도 어울리는 이름은 아니었다. 오늘날 우리는 컴퓨터과학자라고 불린다. 우리가 힘을 합치면 컴퓨터의 위력을 무궁무진하게 개발할 수 있을 것이다”고 말했다.

크누스는 여전히 『컴퓨터 프로그래밍 기술』 제4권을 구상하고 있다.⁶ 그 계획을 구상했던 1962년 첫날 그는 ‘결합 연산’이라는 장을 첨가했었다. 그것은 당시 크누스에게 단지 흥미거리에 불과했지만 지금은 컴퓨터 프로그래밍의 가장 발달된 분야 중의 하나가 되었다. 그는 한 권에 그 주제를 다 실을 수 없을 것이라고 생각했다. 7권의 프로그래밍 시리즈를 끝마친 후에 그는 컴퓨터과학 만큼 중요한 삶의 일부인 작곡을 하고 싶어 한다. 애디슨-웨슬리사의 편집장 피터 고든은 1986년 크누스를 위한 만찬장에 참석해서 “그의 『컴퓨터 프로그래밍 기술』에 나타난 업적은 유클리드에, 그리고 T_EX에 관한 업적은 구텐베르크에 필적하는 것이다. 우리는 그의 이후 활동이 누구에 필적하는 것이 될지 궁금할 뿐이다”고 말했다.

크누스는 남은 프로그래밍 시리즈 4권을 마저 끝내는 데 앞으로 20년의 세월이 필요할 것이라고 생각한다. 아직도 가야 할 길이 너무 많이 남아 있다. 그러나 그는 컴퓨터가 무엇을 위한 것인지 결론을 내려놓고 있다. 그는 컴퓨터의 성격과 기능을 구분하는 멋진 공식을 갖고 있다. 예술과 과학의 차이는, 인간이 컴퓨터로 설명할 수 있을

⁶ 크누스의 홈페이지에 앞으로 나올 책 제목이 모두 공개되어 있다. 그중 제4권은 특이하게도 분책(Fascicle) 2~4편이 2005~2006년 발간되었다.

14 컴퓨터를 과학의 경지로

정도로 충분히 잘 이해하고 있는 것이 과학의 영역이며 그 이외의 것들은 모두 예술의 영역이라는 것이다.