Program
Programming
Programmer

14

CODE 2판

CODE: The Hidden Language of Computer Hardware and Software 2/E

by Charles Petzold

CODE 하드웨어와 소프트웨어에 숨어 있는 언어 2판

전자책 1쇄 발행 2024년 1월 11일 **지은이** 찰스 펫졸드 **옮긴이** 김현규 **펴낸이** 한기성 **펴낸곳** (주)도서출판인사이트 **편집** 백주옥
등록번호 제2002-000049호 **등록일자** 2002년 2월 19일 **주소** 서울특별시 마포구 연남로5길 19-5 **전화** 02-322-5143 **팩스** 02-3143-5579 **블로그** https://blog.insightbook.co.kr **ISBN** 978-89-6626-432-2

CODE 2판

하드웨어와 소프트웨어에 숨어 있는 언어

찰스 펫졸드 지음 | 김현규 옮김

차례

옮긴이의 글

《CODE》가 돌아왔습니다.

프로그래밍과 컴퓨터 공학 분야의 클래식이라 불리는 책들 중 하나인 《CODE》가 그 내용을 개정하고 보강해서 새롭게 출간되었습니다. 1판은 1999년에 나왔고, 2판은 2022년에 나왔으니 대략 23년 만에 새로 출간된 것입니다. (국내 번역서 기준으로는 예전에 한 번 번역서가 나왔으나 절판되었고, 이후에 인사이트 출판사에서 감사하게도 저에게 1판 번역을 맡겨 주셔서 2015년에 1판이 나왔었습니다.)

1판의 '옮긴이의 글'에도 적었지만, 클래식이라는 것은 시간을 넘어설 수 있는 힘을 가진 것을 의미합니다. 그렇다면 어떤 부분 때문에 새로운 판이 나온 걸까요? '2판 서문'에도 나와 있지만, 프로세서의 동작을 설명하고 싶었던 저자의 열망이 컸던 것 같습니다. 개인적으로 10년 넘게 프로세서를 만들었고, 이후에도 그 지식을 바탕으로 영역을 넓혀가면서 엔지니어의 삶을 이어가고 있는 사람으로서, 매우 환영할 만한 결정이라고 생각합니다.

그렇다면 프로세서의 동작을 이해하는 것이 왜 중요할까요? 그리고 많은 사람이 내용을 알고 싶어 하는 걸까요? 프로세서에는 지적 호기심을 자극하는 부분이 있습니다. 이런 재미가 가장 중요한 이유겠지만, 프로세서가 가진 잠재력을 최대한 끌어내기 위해서도 프로세서의 동작에 대한 이해가 필요합니다. 프로세서들은 소프트웨어에서 적절하게 사용될 것이라고 가정하고 만들어지는 경우가 많기 때문이죠. 즉, 운영체제나 컴파일러와 같은 시스템 프로그램은 물론이고, 응용프로그램에서도 프로세서를 더욱 효과적으로 사용할 수 있는 방법들이 존재하며, 이는 프로세서의 동작을 잘 알수록 더 잘 끌

어낼 수 있는 부분입니다.

물론, 소프트웨어 엔지니어 혹은 하드웨어 엔지니어의 길로 들어서려는 초보자들에게 컴퓨터의 동작이란 신비롭게 보일 수도 있습니다. 컴퓨터는 말 그대로 '계산'을 하는 장치에서 기원했으나, '메모리에 저장된 프로그램(stored program)'으로 제어하는 방식이 고안되면서 다양한 작업을 수행할 수 있는 장치가 되었습니다. 즉, 메모리에 저장된 값을 명령으로 인식시킴으로써 메모리에 저장된 값, 즉 프로그램에 따라 다양한 작업을 처리할 수 있는 다재다능한 장치가 되었습니다. 이런 발전 과정을 하나씩 따라가면서 여러분의 지적 호기심을 충족시킬 수 있을 것입니다.

제 생각에 이 책의 가장 중요한 가치는, 초보자 혹은 비전공자라 하더라도 즐겁게 읽어나갈 수 있다는 점입니다. 이 책의 앞부분에서는 코드에 대한 개념을 여러 가지 이야기와 역사적 배경, 다양한 예제를 통해서 풀어냈기 때문에 크게 인식하지 않더라도 자연스럽게 코드와 디지털 논리 회로에 대한 기본적인 개념을 같이 이해할 수 있게 될 것이라 생각합니다. (저도 독자들이 쉽게 읽을 수 있도록 번역하려고 노력했습니다만, 아쉬움이 남습니다. 쉽게 읽히지 않는다면 모두 저의 책임입니다.) 이 부분이 이 책에서 가장 빛나는 부분이라 생각합니다.

책의 뒷부분은 많은 부분에서 보강되고 바뀌었습니다. 1판은 컴퓨터를 구성하는 부분에서 주변부만 다루고 약간 빠르게 끝났습니다만, 2판에서는 프로세서의 기반이 되는 연산기에서 시작해서, 소프트웨어와 하드웨어의 접점이자 가장 중요한 인터페이스인 명령어 셋 아키텍처를 설명하고, 이 명령어들을 이용해서 어떻게 프로세서의 제어와 데이터 처리로 바꿀 수 있는지, 주변 장치와 어떻게 상호작용할 수 있는지를 포괄적으로 다루고 있습니다. 또한 여기서 만든 프로세서 위에서 동작하는 운영체제와 코딩에 이르기까지 기초적인 컴퓨터 아키텍처 책으로 사용해도 될 정도의 내용으로 구성되어 있습니다. 다만, 이 부분은 아주 쉽게 읽히지는 않을 수 있습니다. 워낙 다양하고 정교한 내용을 다루고 있는데, 앞부분의 내용을 알아야 하는 경우가 많기 때

문이죠. 너무 걱정하지 말고 차분히 약간의 인내심을 가지고 읽어보길 권해드립니다. 만일 그래도 이해가 되지 않으면 기술적으로 자세한 부분은 그냥 지나가도 됩니다. 전반적인 흐름을 따라가서 프로세서 하드웨어 위에 소프트웨어가 올라가고, 동작하는 방식을 이해하는 것이 중요하기 때문입니다.

《CODE》는 저에게 아주 특별한 의미가 있는 책입니다. 책의 번역을 다시 맡겨 주시고, 중간 중간 말도 안 되는 실수를 했을 때 꼼꼼히 확인하고 질문을 해 주신 백주옥 편집자님께 감사드립니다. 그리고 이 책의 전체 내용을 검토하고, 어색한 부분과 잘못된 내용을 잡아주신 (주)도서출판인사이트의 한기성 대표님께도 진심으로 감사드립니다. 이 두 분이 없었다면 제가 좋아하는 책을 제 손으로 직접 망치게 되었을 것입니다.

마지막으로, 귀한 주말 시간 동안 게임, 프로그래밍 아니면, 책 읽기와 책 번역을 반복하고 있는 심심한 남편을 늘 지지해 주는 안사람과, 이제는 중학생이 되어 더 이상 아빠의 책을 읽어 주지는 않지만 주말마다 부족한 카페인에 시달리는 아빠에게 커피를 사다 주는 우리 딸에게 감사할 뿐입니다. 감사합니다!

2판 서문

이 책의 초판은 1999년 9월 출간되었는데, 마침내 결코 갱신하지 않아도 되는 책을 썼다는 사실을 깨닫고 매우 기뻐했습니다. 제가 처음으로 쓴 마이크로소프트 윈도우용 응용프로그램을 프로그래밍하는 것에 관한 책은 불과 10년 만에 5번째 판까지 나왔으니, 이 책과는 완전히 비교가 됩니다. 저의 두 번째 책인 OS/2 프레젠테이션 매니저(이게 뭘까요?)에 관한 책은 훨씬 더 빠르게 쓸모 없는 책이 되어 버렸습니다. 하지만 이 책 《CODE》는 영원히 지속될 수 있을 거라 확신했습니다.

이 책에 대해 제가 원래 가지고 있던 생각은, 매우 단순한 개념에서 시작해서서히 디지털 컴퓨터의 동작에 대해 깊이 이해하도록 하는 것이었습니다. 이런 점진적으로 지식의 언덕을 올라가는 과정을 통해서, 은유와 유추, 바보 같은 삽화 등의 사용을 최소화하는 대신 실제 엔지니어들이 사용하는 언어와 기호를 사용할 수 있었습니다. 또한 이 책에서는 나름 재미있는 기법을 통해서 이야기를 풀어나가려 했습니다. 즉, 이 책에서는 보편적인 원리를 설명하기 위해서 고대의 기술을 사용했는데, 고대의 기술이 매우 오래되었음에도 낡은 기술이 아니라는 생각을 가지고 있었기 때문입니다. 마치 내연기관에 관한 책을 쓰면서 포드 모델 T를 기반으로 설명하는 것처럼 말이죠.

저는 이런 접근방식이 여전히 타당하다고 생각하지만, 몇몇 세부적인 부분에서는 틀린 부분도 있었습니다. 시간이 지남에 따라 문화적 요소를 참고한 부분들이 낡은 내용이 되면서, 책이 나이를 먹은 것이 드러나게 되었습니다. 키보드와 마우스를 스마트폰과 손가락이 보완하게 되었으며, 1999년에도 인터넷이 있었지만 지금처럼 될지는 몰랐습니다. 초판에서는 세계의 모든 언

어와 이모티콘을 공통적으로 표현할 수 있는 유니코드에 한 페이지도 채 할애하지 않았으며, 웹에서 가장 많이 사용하고 있는 프로그래밍 언어인 자바스크립트에 대해서는 언급하지도 않았습니다.

이런 문제는 비교적 쉽게 고칠 수 있겠지만, 초판부터 계속해서 저를 괴롭혔던 다른 부분이 있습니다. 바로 컴퓨터의 두뇌, 심장이자 영혼인 CPU가 어떻게 동작하는지 보여 주고 싶었지만, 초판에서는 제대로 보여 주지 못했습니다. 저는 결정적인 돌파구에 거의 도달했다고 느꼈지만 포기하고 말았습니다. 이 점에 대해서 독자들이 불평하지는 않는 것 같았지만, 저에게는 명백한 결함이라는 생각이 들었습니다.

이번 2판에서 이 결함이 수정되었습니다. 이 부분을 추가하면서 70페이지(원서 기준) 정도가 늘어났습니다. 네, 더욱 긴 여정이 되었지만, 이번 2판에서도 저와 같이 페이지를 따라 가다 보면 CPU의 안쪽 깊은 곳까지 도달할 수 있을 거라 생각합니다. 이것이 여러분에게 더욱 즐거운 경험일지 아닐지는 잘 모르겠습니다. 만일 너무 어려워서 질식할 것 같으면 잠시 바람을 쐬러 나갔다 와도 됩니다. 하지만 24장을 통과하고 나면 꽤 뿌듯함을 느낄 것이고, 이 책의 나머지는 식은 죽 먹기라는 것을 알게 되면서 기뻐하게 될 것입니다.

책에 도움이 되는 웹사이트

이 책의 초판에서는 전기의 흐름을 나타내기 위해서 그림의 회로에서 빨간색을 사용했습니다. 이번 2판에서는 CodeHiddenLanguage.com이라는 새로운 웹사이트를 통해서 회로 그림의 내용을 직접 조작해 보고, 회로의 동작을 확인할 수 있도록 하였습니다.

이 책의 전반에 걸쳐 이 웹사이트에 대해서 설명해 두었으며, 몇몇 문단의 여백 부분에 웹사이트를 참고하라는 특별한 아이콘을 표시해 두었습니다. 이후로 이 아이콘을 보면(보통은 회로도 옆에 있는 경우가 많습니다) 웹사이트에서 해당 회로가 어떻게 동작하는지 살펴볼 수 있을 것입니다(기술적인

배경을 알고 싶어 하는 분들을 위해서 밝히자면, 이 웹 그래픽은 HTML5 캔버스 요소를 사용해서 자바스크립트로 프로그래밍했습니다).

CodeHiddenLanguage.com은 완전히 무료로 사용할 수 있습니다. 유료 구독을 해야 볼 수 있는 부분이 있는 것도 아니고, 이 책에 대한 광고 정도만 붙어 있습니다. 몇몇 예제에서 이 웹사이트가 쿠키를 사용하는데, 컴퓨터에 정보를 저장할 수 있도록 허용하는 것일 뿐입니다. 이 웹사이트는 여러분의 활동을 추적하거나 뭔가 나쁜 일을 하지는 않습니다.

저 또한 이 책의 내용을 명확히 하거나 수정하기 위해서 이 웹사이트를 사용할 것입니다.

책임이 있는 사람들

이 책에 대해서 책임을 져야 하는 사람들 중 한 명의 이름이 표지에 나와 있으며, 다른 사람들도 마찬가지로 빠져서는 안 되기 때문에 여기에서 언급하려고 합니다.

특히, 희한하게도 제가 준비된 정확한 시점에 저에게 접근해서 2판을 출간할 생각이 있는지 물어봐 준 편집 담당 이사 헤이즈 험버트Haze Humbert에 대해 먼저 이야기해야 할 것 같습니다. 저는 2021년 1월부터 작업을 시작했는데, 책의 마감 기한이 몇 달 지나고 제가 완전히 한물간 것은 아니라는 약간의 확신이 필요할 때, 그녀는 능숙하게 시련을 통과할 수 있도록 도와주었습니다.

초판의 프로젝트 에디터였던 캐슬린 앳킨스Kathleen Atkins 역시 제가 하려는 일을 잘 이해해 주었으며, 즐겁게 협업을 할 수 있도록 해 주었습니다. 당시 저의 대리인이었던 클로데트 무어Claudette Moore는 제 책의 가치를 알아보고 마이크로소프트 출판사를 설득해서 이 책이 출간될 수 있도록 해 주었습니다.

초판의 기술 편집자였던 짐 푸치스Jim Fuchs는 수많은 당황스러운 오류들을 잡아 주었던 것으로 기억합니다. 이번 2판의 기술 편집자인 마크 시먼Mark Seemann과 래리 오브라이언Larry O'Brien 역시 몇 가지 실수를 찾아 주었고, 이전보다 지면의 내용이 더욱 충실해질 수 있도록 도와주었습니다.

저는 수십 년 전에 '모아서 구성하는compose' 것과 '포함시켜 구성하는comprise' 것의 차이를 알고 있다고 생각했지만 그렇지 않았습니다. 이와 같은 오류들을 바로잡을 수 있었던 것은 교정 편집자인 스카우트 페스타Scout Festa의 가치를 매길 수 없을 정도로 소중한 기여 덕분이었습니다. 저는 항상 교정 편집자의 친절한 도움에 의존해 왔으며, 이들은 익명의 기여자로 남는 경우가 많지만, 부정확함과 언어의 남용에 끊임없이 맞서 싸우는 분들입니다.

이 책에 남아 있는 오류는 전적으로 저의 책임입니다.

셰릴 캔터Sheryl Canter, 잔 이스트룬드Jan Eastlund, 고 피터 골드만Peter Goldeman, 린 마갈스카Lynn Magalska, 그리고 나중에 저의 아내가 된 데어드레 시놋Deirdre Sinnott 등 초판의 베타 리더분들께 다시 한번 감사드립니다.

초판에 실린 수많은 그림은 고 조엘 판초트Joel Panchot의 작품이며, 그는 이 책에 실린 그의 작품에 대해 자부심을 가지고 있었습니다. 그의 그림들 중 많은 작품이 남아 있지만, 회로도가 추가로 필요해지면서 새로운 회로도와의 일관성을 위해서 모든 회로를 새로 그려야 했습니다. (기술적인 배경을 조금 더 설명하면, 여기 있는 그림들은 SVGScalable Vector Graphics 파일을 생성하는 SkiaSharp 그래픽 라이브러리를 사용해서 제가 직접 C#으로 그렸습니다. 수석 콘텐츠 제작자 트레이시 크룸Tracey Croom의 조언에 따라, 이 SVG 파일들은 페이지 조판을 위해 어도비 인디자인Adobe InDesign에서 EPSEncapsulated PostScript 파일 형식으로 변환한 다음 사용했습니다.

그리고 마지막으로

저는 이 책을 제 인생에서 가장 중요한 두 여인께 바치고 싶습니다.

저의 어머니는 소수자를 힘들게 하는 다양한 역경과 싸우셨으며, 결코 저를 막지 않고 언제나 저에게 명확한 방향을 제시해 주셨습니다. 우리는 이 책을 쓰는 동안에 그녀의 95번째(그리고, 마지막) 생일을 축하할 수 있었습니다.

저의 아내 데어드레 시놋을 빼놓을 수 없습니다. 저는 그녀의 업적, 그녀의 지지, 그리고 그녀의 사랑을 끊임없이 자랑스럽게 여깁니다.

그리고 친절한 피드백으로 저를 정말 기쁘게 해 준 초판 독자들에게도 감사드립니다.

2022년 5월 9일

찰스 펫졸드

code (kōd) *n.*

1. a. 메시지 전달에 필요한 문자나 숫자를 나타내기 위하여 사용되는 신호 체계
   b. 비밀리에 혹은 짧게 메시지를 전송하기 위해서 사용되는 약속된 의미를 부여한 기호, 문자 또는 단어의 체계
2. a. 특정 컴퓨터 프로그램을 구성하는 정보
   b. 컴퓨터에서 명령을 표현하기 위하여 사용되는 부호의 표현 체계와 규칙

– The American Heritage Dictionary of the English Language

친한 친구와의 대화

Best Friends

여러분이 10살 소년이라 가정해 봅시다. 가장 친한 친구가 길 건너편에 살고 있으며, 친구의 침실 창문과 여러분의 침실 창문은 서로 마주보고 있습니다. 매일 밤 부모님은 언제나처럼 너무도 이른 시간에 "가서 자라"고 말씀하시곤 하지만, 그 이른 시간에 잠자리에 들기에는 여러분에게는 친구와 나눠야 할 생각과 관찰한 것들, 비밀들, 소문들, 농담, 꿈이 있습니다. 다른 사람과 대화하고자 하는 욕구는 인간의 기본적인 욕망 중 하나이므로 누구도 여러분을 비난할 수는 없을 것입니다.

침실에 아직 불이 켜 있다면, 창문을 통해 친구와 서로 손을 흔든다거나 이런저런 몸짓이나 원초적인 바디 랭귀지를 이용해서 한두 가지 정도의 간단한 생각을 전할 수 있을 겁니다. 하지만 이 방법으로는 복잡한 내용을 전달하기가 매우 어렵고, 부모님이 오셔서 "불 꺼!"라고 말씀하실 테니 이 방법을 더 이상 사용할 수 없을 때의 해결 방법이 필요합니다.

자, 이제 어떻게 친구와 의사소통을 할 수 있을까요? 다행히 휴대폰을 가지고 있는 10살짜리 어린이라면 몰래몰래 전화를 하거나 문자를 사용하면 될 것입니다.

하지만 잠자리에 들 시간에는 부모님이 핸드폰을 사용하지 못하게 한다거나 와이파이Wi-Fi를 꺼버리는 경우에는 어떻게 해야 할까요? 게다가 전자적

통신 방법이 없는 침실은 매우 고립된 방이라 할 수 있습니다.

적어도 여러분과 친구 모두 손전등 정도는 가지고 있을 것입니다. 다들 알고 있겠지만, 손전등이란 물건은 어린이들이 이불 속에서 몰래 책을 보기 위한 용도로 만들어진 것이니까요. 손전등은 어두운 밤에 의사소통을 하는 데 유용하게 사용될 수 있을 것 같습니다. 이 유용한 물건은 매우 조용하고, 빛의 직진성도 뛰어나기 때문에 침실문 밖으로 빛이 새어 나가서 의심 많은 부모의 주의를 끄는 일이 있을 것 같지도 않습니다.

손전등으로 대화가 가능할지 확신은 없지만 한번 시도해 볼 만합니다. 초등학교 때 글자와 단어를 어떻게 적는지는 이미 배웠으니, 그 방법을 손전등으로 적용해 보는 것도 괜찮을 것 같습니다. 바로 창문 앞에 서서 빛으로 글자를 적어 보는 거죠. 예를 들어, 'O'를 적는다고 가정하면 손전등을 켜고 허공에 둥글게 원을 그리고 난 후 손전등을 끄면 되는 거죠. 'I'를 적는다고 가정하면 세로로 한 획을 그으면 되는 것이고요. 하지만 이런 방법이 뜻대로 잘되지는 않는다는 걸 금방 알게 될 것입니다. 친구가 허공에 만드는 곡선과 직선을 보면서, 글자의 획들을 머릿속에서 조합하고 이해하기가 쉽지 않다는 것을 알아채게 될 테니까요. 손전등으로 빙빙 돌리고 내리그어 만드는 빛은 그리 정밀하지 않습니다.

아마도 영화에서 선원들이 바다에서 서로의 배에 불빛을 깜빡거려서 신호를 전달하는 장면을 본 적이 있을 것입니다. 스파이가 거울을 살짝살짝 움직여서 태양빛을 다른 스파이가 사로잡혀 있는 방으로 반사시켜 신호를 보내는 장면이 있는 영화도 있습니다. 이런 방법이 해결책이 될 수 있겠네요. 자, 간단한 기법을 만들어 봅시다. 각각의 알파벳마다 몇 번씩 빛을 깜빡거릴지 정하는 겁니다. A는 1번, C는 3번, Z는 26번 빛을 깜빡거리는 거죠. BAD라는 단어를 전달하려면 2번, 1번, 4번을 깜빡거리면 되겠습니다. 물론, 친구가 7번 깜빡거린 걸로 인식해서 G로 이해하지 않도록 하기 위해서 글자와 글자 사이에 어느 정도 시간을 두는 것이 필요하겠지요. 단어 사이에는 조금 더 시간을 두면 될 것 같네요.

이 방법은 괜찮을 것 같습니다. 좋은 소식은 이제부터는 손전등을 친구의 창문에 비추고 스위치만 껐다 켰다 하면 되므로 더 이상 손전등을 허공에서 흔들고 있지 않아도 된다는 것이지요. 나쁜 소식은 아마도 첫 번째로 보내려고 하는 'How are you?'라는 메시지 하나에 손전등을 총 131번 깜빡거려야 한다는 거죠. 게다가 문장부호에 대해서는 생각해 보지 않았으니, 물음표를 표현하기 위해서 몇 번을 깜빡여야 할지 모른다는 점도 있지요.

그래도 많이 접근했습니다. 아마 다른 사람들도 분명히 이 문제에 직면했을 거라고 생각되지 않으세요? 네, 맞습니다. 이 문제에 대해서 알아보기 위해 도서관이나 인터넷을 찾아보면, 모스 부호Morse code라 불리는 놀라운 발명을 찾을 수 있을 것입니다. 이것은 여러분이 찾고 있던 바로 그 해결책이지요. 비록 알파벳의 모든 글자를 어떻게 '적어야' 하는지 다시 배워야 한다는 문제가 남아 있지만 말입니다.

차이점은 다음과 같습니다. 여러분이 만든 체계에서는 모든 알파벳에 대하여 A는 1번, Z는 26번과 같은 방식으로 깜빡임의 수를 정해 두었지만, 모스 부호에서는 짧은 깜빡임과 긴 깜빡임이라는 두 가지 형태의 깜빡임을 가지고 있습니다. 이 부분이 모스 부호를 좀더 복잡하지만, 좀더 효율적으로 만들어 주는 것입니다. 모스 부호에서는 'How are you?'라는 문장을 표현하기 위하여 131번이 아닌 32번의 깜빡임(몇 개는 짧게, 몇 개는 길게 깜빡여야 하겠지만요)만 필요하며, 물음표도 보낼 수 있습니다.

모스 부호가 어떻게 동작하는지 설명할 때는 '짧은 깜빡임'과 '긴 깜빡임'이라 표현하는 대신 '점dot'과 '선dash'이라 부릅니다. 이는 '점'과 '선'이 종이 위에 모스 부호를 표현하는 가장 일반적인 방법이기 때문이지요. 모스 부호에서는 알파벳의 모든 문자를 점과 선으로 된 짧은 조합으로 표현하고 있으며, 이는 다음 표와 같습니다.

A	•–
B	–•••
C	–•–•
D	–••
E	•
F	••–•
G	––•

H	••••
I	••
J	•–––
K	–•–
L	•–••
M	––
N	–•

O	–––
P	•––•
Q	––•–
R	•–•
S	•••
T	–
U	••–

V	•••–
W	•––
X	–••–
Y	–•––
Z	––••

비록 모스 부호가 컴퓨터로 할 수 있는 일에 비할 바는 아니지만 이를 통하여 부호(코드/code)의 속성에 익숙해지는 것이 컴퓨터 하드웨어와 소프트웨어에 숨겨진 언어와 내부 구조를 좀더 깊이 이해하는 데 반드시 필요합니다.

이 책에서 **부호**(코드/code)라는 용어는 일반적으로 사람 사이에, 사람과 컴퓨터 간에, 그리고 컴퓨터끼리 정보를 전달하는 체계를 의미합니다.

부호는 의사소통을 할 수 있게 해 줍니다. 간혹 부호는 암호의 형태를 가지지만, 대부분은 그렇지 않습니다. 대부분의 부호가 인간 의사소통의 기반으로 사용되기 위해서는 이해하기 쉬워야만 하기 때문이죠.

우리가 단어를 만들기 위해서 입으로 만들어 내는 소리는, 이 목소리를 들을 수 있으며 이 언어를 알고 있는 누구나 이해할 수 있는 부호로 구성되어 있습니다. 이런 부호를 '음성 언어' 혹은 '말'이라 부릅니다.

청각 장애인 공동체 내에서 사용하는 다양한 수어는 손과 팔의 움직임과 형태를 통해서 단어의 각 문자, 단어 자체 혹은 개념을 전달할 수 있습니다. 북미에서 가장 일반적으로 사용되는 두 가지 수어 시스템은 19세기 초 미국 농인 학교에서 개발된 미국 수화American Sign Language, ASL와 프랑스 수어의 변형인 LSQLangue des signes Québécoise입니다.[1]

종이나 다른 매체에 쓰는 또 다른 부호가 있는데, 이를 '문자 언어' 혹은 '글'이라 부릅니다. 문자는 손으로 쓰거나 키를 눌러 입력하고 나서 신문, 잡지, 책으로 인쇄하거나 다양한 화면 표시장치에 디지털 형식으로 표시할 수 있

1 (옮긴이) 참고로 한국의 수어는 한국 청각 장애인을 위해서 공용어로 지정되어 있습니다. 지(指)문자를 통해서 각각의 문자를 표현할 수 있으며, 수어를 통해 단어와 개념을 나타낼 수도 있습니다.

습니다. 대부분의 언어에서 말과 글은 매우 강한 연관성을 지니고 있습니다. 예를 들어, 영어에서는 문자와 문자 집합(글)은 음성(말)과 많든 적든 연관성을 가지고 있습니다.

보지 못하는 사람을 위해서 글은 점자로 대체될 수 있습니다. 점자는 튀어나온 점을 이용해서 문자, 문자의 묶음 혹은 단어 전체를 표현하는 체계를 이용하는 부호입니다(점자에 대해서는 3장에 자세히 다루겠습니다).

말을 글로 매우 빠르게 적어야 하는 경우에는 속기술stenography 혹은 약기술shorthand이라 불리는 방식이 유용합니다. 속기사들은 법정이나 TV 뉴스 혹은 스포츠 프로그램에서 실시간으로 캡션을 만들기 위해서 속기 기계를 사용하는데, 이 장치는 속기에 사용되는 특유의 문자에 대응되는 간략화된 키보드를 가지고 있습니다.

일반적으로 의사소통을 위해서 수많은 부호들이 사용됩니다. 이는 몇몇 부호들이 다른 것들보다 특정 측면에서 적합하기 때문이지요. 말은 종이에 저장할 수 없기 때문에 대신 글을 사용하는 것입니다. 어둠에서 소리를 내지 않고 거리가 멀리 떨어진 위치에서 말이나 글로 의사소통을 하는 것은 불가능합니다. 이러한 경우에는 모스 부호가 적절한 대안이 되는 것이지요. 즉, 어떤 부호가 다른 부호로는 처리가 불가능한 경우에도 적절히 사용될 수 있다면 그 부호는 유용하다고 말할 수 있습니다.

마찬가지로 컴퓨터에서도 문자, 숫자, 소리, 음악, 그림, 영화 등을 저장하고 통신하며, 컴퓨터 자체에 명령을 내리기 위해서 매우 다양한 부호가 사용됩니다. 컴퓨터는 인간이 지닌 눈, 귀, 입, 손가락과 같은 기관을 정확히 복제해서 가지고 있을 수 없기 때문에, 인간이 사용하는 부호를 쉽게 사용하기 어렵습니다. 컴퓨터가 말할 수 있도록 가르치는 것은 어렵고, 말을 이해할 수 있도록 만드는 것은 훨씬 더 어렵습니다.

하지만 많은 진전이 있었습니다. 이제 컴퓨터는 시각(문자와 그림), 청각(언어, 소리와 음악), 혹은 두 가지의 조합(애니메이션과 영화)을 포함해서 인간의 대화에 사용되는 다양한 유형의 정보를 취득하고, 저장하고, 조작하

고 만들 수 있게 되었습니다. 이러한 모든 유형의 정보를 다루기 위해서는 자체적인 부호가 필요합니다.

앞에서 보았던 모스 부호 표도 그 자체로 일종의 부호입니다. 이 표에서는 각 문자를 표현하기 위한 점과 선의 조합을 표시했지만, 실제로 점과 선을 보내는 것은 아니지요. 손전등으로 모스 부호를 전달할 때 단지 점과 선은 깜빡임의 종류에 대응하는 것입니다.

손전등으로 모스 부호를 보낼 때, 전등 스위치를 매우 빠르게 점멸시켜서 빠른 깜빡임을 만들면 이는 점에 해당합니다. 손전등을 좀더 오랫동안 켜 두면 느린 깜빡임이 만들어지며, 이는 선에 해당합니다. 예를 들어, 'A'를 보낸다고 하면, 손전등의 불빛을 매우 빠르게 한 번 점멸시킨 후 좀더 느리게 점멸시키면 됩니다. 다음 글자를 보내기 전에는 잠시 시간을 가져야겠죠. 일반적으로 선을 표현할 때는 점을 표현할 때보다 약 3배 정도 오랫동안 불빛을 켜 둡니다. 예를 들어, 점이 1초 길이라면 선은 3초 길이가 되는 것이지요(물론 실제 모스 부호를 전송할 때는 이것보다 훨씬 빠르게 진행됩니다). 수신자는 짧은 깜빡임을 보고, 긴 깜빡임을 본 후 어느 정도 불이 꺼져 있다면 이것이 'A'라는 걸 알 수 있습니다.

모스 부호에서 점과 선 사이에 잠깐 시간을 두는 것은 반드시 필요합니다. 예를 들어, 'A'를 보내는 경우 점과 선 사이에 손전등을 반드시 꺼야 하고, 이 길이는 하나의 점을 표현하는 시간과 동일합니다. 각 글자 사이에서도 손전등을 꺼서 구분하게 되는데, 이 간격은 하나의 선을 표현하는 시간과 동일합니다. 다음 그림은 'hello'를 표현하는 모스 부호에서 문자 간의 간격을 표시한 것입니다.

단어 간에는 선 두 개를 표현하는 정도의 시간을 띄웁니다. 다음은 'hi there'를 표현하고 있습니다.

손전등을 켜거나 꺼두는 시간은 정해져 있지 않습니다. 이 시간은 모두 점을 표현하는 시간을 기준으로 상대적으로 표현되는 것이며, 이 시간이 얼마나 걸릴지는 손전등을 얼마나 빠르게 깜빡일 수 있는지와 보내는 사람이 얼마나 빨리 특정 글자에 대한 모스 부호를 기억해 내는지에 달려 있습니다. 빨리 보내는 사람이 선을 표현하는 시간과 느리게 보내는 사람이 점을 표현하는 시간이 같을 수도 있다는 것이지요. 사소한(?) 문제는 모스 부호를 받는 사람이 부호를 읽어내는 것이 어렵다는 것인데, 그래도 한두 개 정도의 문자를 받고 나면 받는 사람이 보통은 어떤 것이 점이며, 어떤 것이 선인지를 구분할 수 있습니다.

얼핏 모스 부호에서 각 알파벳 문자에 점과 선을 이용한 조합을 할당한 게 키보드에서 그러하듯이 규칙 없이 이루어진 것처럼 보입니다. 그러나 좀더 꼼꼼히 살펴보면, 완전히 그런 것만은 아니라는 사실을 알 수 있습니다. 영어에서 자주 사용되는 E와 T 같은 문자에는 좀더 짧고 간단한 부호가 할당되어 있죠. 낱말 맞추기 게임을 좋아하는 분들이라면 이런 방법이 옳다는 걸 알 수 있을 것입니다. 일반적으로 사용되지 않는 Q와 Z 같은 문자들의 경우 좀더 긴 부호가 할당되어 있습니다.

여러분들은 대부분 모스 부호에 대하여 아주 조금은 알 거라 생각합니다. 세 번의 점, 세 번의 선, 세 번의 점은 'SOS'로 국제 조난 신호를 나타내죠. 'SOS'는 어떤 단어의 약어가 아니고, 단지 쉽게 기억할 수 있는 모스 부호 조합을 선택한 것입니다. 제2차 세계대전 중 영국의 BBC에서는 몇몇 방송을 시작할 때 베토벤 교향곡 5번 운명을 자주 틀곤 했는데, 이는 (베토벤이 작곡할 때는 전혀 알지 못했겠지만) 운명 교향곡의 도입부인 '바바바바~~~암' 하는 부분을 모스 부호로 나타내면 승리victory를 의미하는 'V'가 되기 때문입니다.

모스 부호의 단점은 대문자와 소문자의 구분이 없다는 점입니다. 하지만 모스 부호는 문자뿐만 아니라 5개의 점과 선을 조합해서 숫자도 표현할 수 있습니다.

1	•▬▬▬▬
2	••▬▬▬
3	•••▬▬
4	••••▬
5	•••••

6	▬••••
7	▬▬•••
8	▬▬▬••
9	▬▬▬▬•
0	▬▬▬▬▬

이러한 숫자 부호들은 적어도 영문자보다는 순서 있게 구성되어 있습니다. 대부분의 문장부호는 5개, 6개 혹은 7개의 점과 선의 조합으로 이루어져 있습니다.

.	•▬•▬•▬
,	▬▬••▬▬
?	••▬▬••
:	▬▬▬•••
;	▬•▬•▬•
-	▬••••▬
/	▬••▬•
"	•▬••▬•

'	•▬▬▬▬•
(	▬•▬▬•
)	▬•▬▬•▬
=	▬•••▬
+	•▬•▬•
$	•••▬••▬
¶	•▬•▬••
–	••▬▬•▬

모스 부호에는 유럽의 언어들을 위한 악센트 부호가 있는 문자와, 특수한 목적으로 사용하기 위한 속기 문자 조합과 같은 추가 부호들도 정의되어 있습니다. 'SOS'와 같은 것이 이러한 속기 부호의 예가 될 수 있으며, 이 부호를 전달할 때는 세 문자가 전송됨에도 문자 간에 점 하나 정도의 시간 간격만을 가지게 됩니다.

만일 모스 부호를 전달하기 쉽도록 만들어진 손전등을 가지고 있다면, 모스 부호 전달이 훨씬 쉽겠지요. 일반적인 손전등과 다르게 이 손전등은 손전등을 켜고 끄는 스위치 이외에 버튼이 하나 더 있어서, 버튼을 누르고 떼는 동작에 따라서 불이 켜지고 꺼질 수 있게 만들어져 있습니다(보통 군용 전등이라고 이야기하는 손전등이 이렇게 생겼습니다). 조금만 연습을 하면 1분에 5~10단어 정도를 보내고 받을 수 있을 것입니다. 분당 100단어 정도를 전달할 수 있는 말하기에 비해서는 매우 느리지만, 이 정도면 충분히 의사소통에

활용할 수 있을 정도는 됩니다.

모스 부호를 완전히 외우고 난 이후에는(모스 부호를 좀더 익숙하게 보내고 받기 위해서는 외울 수밖에 없겠죠?) 모스 부호를 소리로 내어 통상적인 말을 대체할 수도 있습니다. 속도를 올리기 위해서 점dot은 '디'라고 부르고(마지막 점인 경우엔 '딧'이라고 부릅시다), 선dash은 '다'라고 부릅시다(예를 들어, V를 나타내는 모스 부호는 '디-디-디-다'로 읽을 수 있습니다.). 이러면 모스 부호를 적을 때는 점과 선으로, 이걸 읽을 때는 단 두 가지 모음으로 줄일 수 있습니다.

여기서 가장 중요한 단어는 '두 가지'입니다. 두 가지 형태의 깜빡임, 두 가지 모음, 실제로 두 가지 다른 것들은 적절히 조합되어 모든 형태의 정보를 전달할 수 있습니다.