

## <컴퓨터의 눈부신 발전, 그리고 한계>

원종훈

부울 대수와 스위치에 대해 공부하면서 새년이라는 인물이 존재했다는 것을 처음 알게 되었는데, 그의 업적이 컴퓨터의 발전에 엄청난 파급효과를 불러일으켰을 것이라고 생각된다. 컴퓨터의 역사에 대한 상식적인 차원으로, 부울, 튜링, 폰 노이만 정도의 이름은 일반인들에게도 어느 정도 친숙하다. 부울의 수학 체계는 컴퓨터과학에서뿐만 아니라 수학 교과서에서도 등장하는 개념이며, 튜링과 폰 노이만은 컴퓨터의 아버지와 같은 사람들이기 때문이다. 그런데, 부울 대수를 스위치의 개념과 연결하여 디지털 논리 회로의 기반을 만든 새년의 공로는 충분히 인식되지 못하고 있는 듯하다. 그는 부울 논리를 스위치 회로에 그대로 적용하며, 임의의 부울 논리식에 해당하는 스위치 회로가 있고, 임의의 스위치 회로에 해당하는 부울 논리식이 있다는 것을 증명함으로써 스위치를 이용한 설계가 체계적으로 진행될 수 있도록 그 기틀을 마련했다. 스위치와 부울 대수를 정확히 대응시킨 그의 석사 논문이 있었기에 오늘 날의 컴퓨터 과학 기술이 존재할 수 있었던 것이 아닐까 싶다.

두 번째로, 컴퓨터 프로그램과 수학이 밀접한 관련이 있다고 배우긴 했지만, 프로그램의 한 줄과 수학적 증명의 한 줄이 일대일 대응이 될 정도로 서로 거울 같은 관계를 가진다는 사실이 굉장히 놀라웠다. 사실 자료형, 함수 등의 기본적인 개념들만 잘 숙지하면 프로그래밍에서 수학에 대한 이해를 크게 필요로 하지 않는다고 흔히들 생각하곤 하는데, 이는 완전히 틀린 말이라는 사실을 이번 기회를 통해 확실히 깨달았다. 수학의 증명에서 사용되는 수많은 논리적 추론의 징검다리들이 실질적으로 프로그램의 한 줄 한 줄과 어떻게 연결되는지를 학습을 통해 명확하게 확인할 수 있었다. 증명 문제를 잘 풀면 자연스럽게 좋은 프로그래머가 될 수 있는 기본적인 자질을 갖추게 되는 것인가 하는 궁금증이 생기기도 했다.

또한, 우후죽순으로 생겨난 수많은 프로그래밍 언어들이 나름대로 공통적인 뿌리와 체계를 통해 분류된다는 사실이 흥미롭게 다가왔다. 요즘 프로그래밍을 배운다고 하면 흔히 C, C++, JAVA, python 등의 대중적인 언어를 가장 먼저 접하게 된다. 나도 역시 기초적인 C와 JAVA 프로그래밍을 공부해본 적이 있는데, 이런 유형의 언어들 외에 다른 종류의 언어들이 존재한다는 것을 전혀 알지 못했다. 그런데, C나 JAVA는 프로그래밍 언어의 큰 두 분류 중 한 갈래인 '기계 중심을 유도하는' 언어에 속해있고, '함수 중심을 유도하는' 유형의, 들어보지도 못한 수많은 언어들이 즐비하다는 것

을 알게 되었다. 이를 보며, 앞으로 프로그래밍을 제대로 공부하거나 코드를 짜야 할 일이 생기게 된다면 자신의 필요와 업무의 성질에 맞는 언어를 잘 택해야겠다는 생각이 들었다. 형식, 기능 등에 따라 선택할 수 있는 수없이 많은 대안이 체계적으로 분류되어 있는데 단순히 많은 사람들이 사용한다는 이유만으로 작업할 언어를 선불리 선택하는 것은 어리석은 판단일 수 있겠다는 생각을 하게 되었다.

디지털 논리회로와 차곡차곡 쌓기(abstraction hierarchy)의 개념에 대한 학습을 하고 나니, CPU나 메모리의 내부 구조가 어떤 식으로 구성되어 있을지 ‘상상’ 정도는 해볼 수 있게 되었다. 이 수업을 듣기 전까지는 컴퓨터 내의 CPU나 메모리 등이 어떻게 생겼을지 감히 상상할 엄두조차 내지 못했다. 막연하게 “굉장히 복잡한 무언가가 있겠지” 하는 생각만 한 채 번번이 진입장벽에 부딪혔다. 그러나 이번 기회를 통해 부울 대수와 스위치를 바탕으로 한 응답회로, 플립플롭 등의 기초적인 디지털 논리 회로를 습득하고, 차곡차곡 쌓기(abstraction hierarchy)의 개념을 익히며 아주 기초적인 이론적 핵심 구조는 어느 정도 알게 되었고, 이들이 어떤 방식으로 연결되어 있을 지를 조금이나마 상상해볼 수 있게 되었다.

마지막으로, 앞으로 일상생활에서 어떤 문제를 접할 때 이 문제가 현실적인 비용의 문제인지, 비현실적인 비용의 문제인지, 혹은 애초에 불가능한 영역의 문제인지를 가늠하다면 머릿속으로 상상을 해보아야겠다는 생각을 갖게 되었다. 비록 우리의 두뇌가 컴퓨터는 아니지만, 나름대로 문제해결 방식의 복잡도를 염두에 두며 일상생활이나 업무에서 맞닥뜨린 문제를 해결하는 데에 얼마나 많은 시간과 노력을 들여야 하는지, 혹은 아예 푸는 것이 불가능한 문제인 것은 아닌지를 어느 정도 미리 판단할 수 있다면, 훨씬 빠르고 효율적으로 문제를 해결하고 의사결정을 내릴 수 있겠다는 생각이 들었다.