

국내 컴퓨팅 교육의 현황, 문제점, 그리고 개선방향

전북대학교 | 오일석* · 양재동**

전남대학교 | 김수형*

1. 컴퓨터 프로그래밍의 국가적 중요성

정부는 2006년 들어 “IT 강국에서 소프트웨어 강국으로”라는 슬로건을 내걸었다. 2006년 7월에는 NSF (National Software Forum) 2006을 개최하여 미래 성장 산업으로 소프트웨어의 중요성을 재삼 확인하였다[1]. 한국정보과학회는 “산업을 변화시키는 소프트웨어 기술”이라는 주제를 특집으로 다룬 학회지를 2007년 2월에 발간하여 시대 요구를 담아내고 있다[2].

각종 하드웨어 장치에 들어가는 소프트웨어 양은 급속히 증가하고 있다. 핸드폰에서 하드웨어 기능의 확장은 더 이상 중요한 자리를 차지하지 못한다. 간단한 일정 관리 프로그램부터 자이로 센서에 의한 첨단 모션 인식 기능까지 그 진화 속도는 매우 빠르며 이러한 속도는 소프트웨어가 좌우하고 있다[3]. F-22 전투기는 전체 기능 중에 80%를 소프트웨어가 담당하고 있으며, 내장된 소프트웨어의 분량은 1000만 라인 이상이라고 알려져 있다.

이러한 상황에서 정부가 소프트웨어에 주목한 것은 매우 바람직하다 할 수 있다. 하지만 여러 대학에서 컴퓨터 소프트웨어 관련 학과에 대한 학생들의 선호도가 급속히 떨어지고 있다. 또한 고등학생들도 컴퓨터를 학문으로 보지 않고 누구나 하는 것으로 여기는 경향이 뚜렷해져 양질의 학생이 신입생으로 유입되지 못하는 실정이다[4]. 또한 대학 컴퓨터 학과의 교과과정 운영도 기업이 요구하는 수준에 크게 미치지 못하고 있어 문제점으로 자리 잡고 있다.

이 글에서는 컴퓨터 프로그래밍 교육의 내용, 프로그래밍을 기피하는 상황과 이유, 그리고 프로그래밍 교육과정 사례, 그리고 상황 타개를 위해 취해야 하는 방안을 제시한다.

2. 프로그래밍과 소프트웨어 개발 교육 내용

* 종신회원

** 정회원

최근 하드웨어의 발전, 멀티미디어 환경의 보편화, 인터넷과 웹의 대중화, 플랫폼의 다양화 등으로 인해 프로그래밍과 소프트웨어 개발에 대한 요구가 크게 달라지고 있다. 이 장에서는 대학에서 꼭 다루어야 할 내용과 교육 방식에 대해 간략히 기술한다.

2.1 알고리즘

알고리즘은 두 가지 측면에서 매우 중요하게 다루어야 한다. 첫 번째는 프로그램의 효율과 관련이 있다. 학생들이 효율의 중요성을 이해할 수 있도록, 프로그래밍 숙제에서는 충분히 큰 자료를 주어야 한다. 예를 들어 탐색(search) 문제의 경우 수백만 내지 수천만 개의 자료를 가진 파일을 제공하여 N이 충분히 큰 경우 서로 다른 알고리즘의 효율을 비교하도록 하여 알고리즘 효율의 중요성을 직관적으로 느끼게 해야 한다. 특히 요즘은 현장에서 방대한 자료를 다루어야 하는 상황이 자주 발생하므로 그 중요성이 더욱 커졌다고 봐야 한다. 두 번째는 알고리즘 교육을 통해 학생들의 지적 호기심을 유발할 수 있다는 점이다. 컴퓨터 프로그래밍이 단순한 제품 제작을 넘어 과학의 한 영역임을 이해할 수 있도록 유도해야 한다.

알고리즘을 이론 과목으로 보고 프로그래밍을 소홀히 다루는 경우도 있는데, 그건 틀린 생각이다. 단순한 배열을 사용하는 프로그래밍 수준을 넘어, 그래프, 트리, 연결 리스트 등에서의 연산을 자유자재로 설계 구현할 수 있어야 한다. 또한 이러한 자료 구조를 어느 상황에서 사용해야 효과적인지를 판단할 수 있는 능력을 갖추기에 충분한 정도의 프로그래밍 숙제를 부과해야만 한다.

2.2 소프트웨어 공학

졸업할 때까지 컴퓨터 프로그래밍은 수천 줄의 프로그램을 혼자 작성하는 것으로 여기는 학생이 적지 않다. 건축학에서 수 십층의 거대 건물을 짓는 상황이 자주 발생하는 것처럼 소프트웨어에서도 수십만~수

백만 줄의 소프트웨어 작성이 빈번히 발생함을 주시시켜야 한다. 특히, 앞서 언급한 전투기 내장 소프트웨어 경우처럼 소프트웨어 라인 수가 1000만 라인이 넘어갈 경우 증가하는 천문학적인 시스템의 복잡도는 주먹구구식의 프로그래밍이 얼마나 무모한 일인가를 단적으로 말해준다. 따라서 소프트웨어 공학이 이러한 복잡도에 체계적으로 대처하기 위한 필수적인 방법론임을 학생들에게 여러 사례 연구를 통해 피부로 느끼게 해야 한다. 문제를 해결하기 위해 바로 코딩에 들어가는 습관을 지양시키며 반드시 요구분석, 설계, 구현, 검사, 유지보수, 비용 산정 등의 과정을 소프트웨어 개발 도구를 사용하여 충실히 따르도록 훈련시켜야 한다. 국내에서는 이 과정이 실제 프로젝트를 거치지 않고 단순 이론에 치우친 강의로 끝나는 경우가 적지 않은데, 이러한 강의 방식은 나무만을 보고 숲을 보지 못하는 공학도만을 양산하는 우를 범하게 된다. 따라서 이러한 방식은 반드시 탈피해야 한다. 다음은 구체적인 코스의 내용을 기술하고 있다.

- 소프트웨어 질의 기준에 대한 명확한 인식
- 소프트웨어의 체계적인 과정을 통한 개발 방법론
- 버전관리, 형상관리 그리고 소프트웨어 테스트 검증 과정등을 적절한 도구 기반으로 체득
- UML(Unified Modeling Language) 도구 기반 객체 지향 소프트웨어 개발 기법 체득
- 모듈이나 컴포넌트 기반의 소프트웨어 개발 기법 습득
- SOA(Service Oriented Architecture), 웹서비스를 위한 시스템 아키텍처, 미들웨어 설계 등 시스템의 골격에 대한 설계 및 구현 과정을 훈련

2.3 웹 프로그래밍

요즈음 소프트웨어 분야에서의 주요한 흐름은 공유, 참여, 개방을 표방하는 웹 2.0의 철학이다. 어찌보면 이 철학은 소프트웨어 분야 뿐 아니라 사회, 경제, 경영, 문화 등 우리 사회의 전 분야로 파급되어 가는 느낌이다. 다른 한편으로, 소프트웨어 분야의 최근 변화를 살펴보면, 시장이 예전의 소프트웨어 솔루션 중심에서 확연히 서비스 중심으로 옮겨간 것을 알 수 있다. 코스닥 시장을 보더라도 소프트웨어 솔루션 중심의 소프트웨어 회사들은 거의 찾아보기 어렵거나 여건이 어려운 반면, NHN, 다음, 엠파스, SK컴즈 등은 성장세에 놓여 있다. 솔루션 전문 회사는 100억 매출 올리기도 어려운 반면 NHN의 자산이 7조이고, 1년 매출을 1조원으로 잡고 있는 것을 보더라도 쉽게 알 수 있다. 외국의 경우를 보더라도 서비스 중심의 구글이

프는 별이라면, 솔루션 중심의 MS사는 지는 해로 비유될 수도 있을 정도이다. 이러한 추세에 부응하는 웹 서비스 관련 프로그래밍이 한 과정으로 추가된다면, 현 소프트웨어 교육의 상황 타개에도 상당한 도움이 될 수 있을 것이다. 다음은 구체적인 프로그래밍 코스 내용이다.

- 오픈 API에 기반한 매시업 서비스 관련 프로그래밍(SOA, REST 프로토콜, XML-RPC, 웹 서비스 프로그래밍)
- 스크립트 기반 웹 프로그래밍(AJAX, 닷넷 프로그래밍, ASP, PHP, JSP 등의 스크립트 언어 프로그래밍)
- 위젯 프로그래밍
- 분산 환경을 지원하는 에이전트 기반 프로그래밍

2.4 다양한 플랫폼에서의 프로그래밍

이제 전통적인 데스크탑 환경에서 작동하는 소프트웨어를 벗어나 다양한 플랫폼에서의 프로그래밍이 갈수록 중요해 지고 있다. PDA, 핸드폰, 텔레매틱스 단말기 등과 같이 소형 메모리와 저성능 프로세서를 탑재한 모바일 기기에서 효율적인 프로그래밍에 대한 요구가 갈수록 커지고 있다[5]. 또한 임베디드 소프트웨어에 대한 교육도 강화해나가야만 한다[3].

3. 컴퓨터 프로그래밍 기피의 상황과 이유

국가가 필요로 하는 고급 소프트웨어 인력 규모를 감안할 때, 현재 학생들의 프로그래밍 기피 현상은 국가적으로 우려스러운 수준이다[6]. 이 장에서는 국내 네 개 대학의 컴퓨터 관련 학과의 최근 선호도 통계를 제시한다. 또한 학생들이 프로그래밍을 기피하는 원인을 분석한다.

3.1 국내의 현황

현재 국내 대학에서 컴퓨터전공에 대한 기피현상이 공통적으로 나타나고 있다. KAIST의 경우 학부생의 학과 선택 추이는 그림 1에서와 같이 2001년을 기준으로 급격히 감소하고 있다. 전북대학교 전자정보공학부와 전남대학교 전자컴퓨터공학부의 경우에도 학생들의 전자나 전기 선호 현상이 두드러지고 있지만, 학생배정 규정 등을 통해 일정수의 학생이 컴퓨터 전공에 배정 보장되도록 운영하고 있다(표 1과 표 2 참조). 컴퓨터학과와 전파공학과가 80:40명의 비율로 구성된 고려대학교 정보통신대학의 경우에도 학생들의 전파공학 선호도가 두드러지고 있다(표 3 참조).

KAIST 학부생 학과 선택 추이

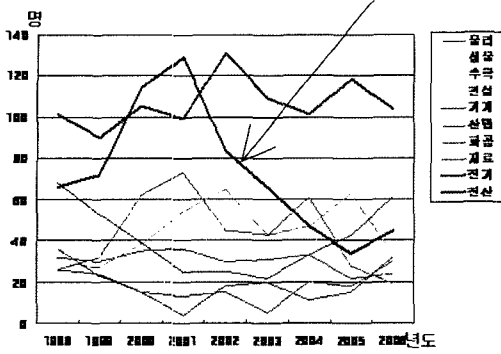


그림 1 KAIST 학부생의 학과선택 추이

표 1 전북대학교 전자정보공학부 전공배정 현황

	지원자수					배정자수		
	전체	컴공	전자	전기	미신청	컴공	전자	전기
2005학번	365	62	208	70	25	108	198	59
2006학번	345	42	224	65	14	115	179	51

표 2 전남대학교 전자컴퓨터공학부 전공배정 현황

	지원자수		배정자수	
	컴공	전자	컴공	전자
2005학번	50	118	95	90
2006학번	76	166	135	134

표 3 고려대학교 정보통신대학의 전공배정 현황

	배정자수	
	컴퓨터	전파
2004학번	69	55
2005학번	71	55
2006학번	72	53

이러한 컴퓨터 기피 현상의 주요 원인은 다음과 같다. 첫째, 중고교에서의 컴퓨터 교육이 한글이나 엑셀, 파워포인트 등에 대한 사용법 위주로 교육을 하고 있기 때문에 중고교 때부터 컴퓨터공학을 학문으로 보지 않고, 단지 아무나 사용할 수 있고 누구나 다 알고 있는 것으로 인식하는 경향이 있다는 점이다.

둘째, 대학 1학년때의 프로그래밍 기피현상을 들 수 있다. 전자 컴퓨터 분야로 입학한 경우로 한정하여 살펴보면 전자는 중고교 때 공부하던 방식의 수식 외우고 풀고 점수 받는 방식으로 운영되는 반면, 프로그래밍은 무언가를 스스로 만드는 것을 요구하니 힘들어하는 학생이 적지 않다. 심지어 학생들 간에 떠도는 이야기 중의 하나는 “컴퓨터를 하고는 싶은데 어렵다더라”이다.

셋째, 프로그래머에 대한 낮은 사회적 인식이 컴퓨터 기피현상으로 이어지고 있다는 점이다. 컴퓨터 분야 종사자는 박봉에 초과 근무까지 하는 것이 일반적이어서 안정적인 직업으로 인정받지 못하고 있다. 이러한 현상의 원인은 소프트웨어 불법 복제와 지적 재산에 대한 낮은 인식 등에 의해 소프트웨어 산업이 제대로 발전하지 못한 탓을 들 수 있다.

넷째, 컴퓨터공학이라는 학문의 경계가 점점 애매해져 가고 있다는 점이다. 특히 전자공학과 비교하여 졸업생 취업, 산업 현황, 성공한 졸업생(대기업 CEO), 교수 연구업적, 산학협동/연구비 실적, 강의평가 등 모든 항목에서 컴퓨터학도가 전자공학과를 못 당하고 있다. 배출한 학생들의 프로그래밍 능력도 학원에서 공부한 사람과 큰 차이를 보이지 못한다.

3.2 중국의 현황

IT산업이 호황을 누리고 있는 중국의 경우 지난 50여 년간 Computer Science and Technology(CST)라는 이름의 전공이 꾸준히 발전하여 2005년말 현재 중국내 대학에서 가장 큰 전공으로 자리매김하고 있다. 중국내에서 이공계 학과를 가진 644개 정규대학 중 505개 대학에 CST 전공이 개설되어 있으며, 최근 5년간 30만명 내외의 학생이 입학하였다[7].

이렇듯 잘 나가는 중국에서도 IT분야 학생수가 최근 적지 않게 감소하고 있다. 통계에 의하면 2001년 중국내 모든 대학의 CST 전공 입학생수는 85,736명이었는데, 2002년에는 77,640명, 2003년에는 63,720명으로 매년 10~18% 가량 감소하고 있다. 이에 따라 2004년부터 정부 주도의 위원회가 구성되어 IT분야 공급학생 감소 현상의 원인분석과 대책마련을 위해 부심하고 있다. 이 위원회가 진단한 IT분야의 문제점은 다음의 세 가지로 요약된다.

- 1) 기업체에서 신규 채용한 인력의 재교육 기간이 점점 길어지고 있다.
- 2) IT 산업에 종사하는 인력은 대학시절 현재 업무에 필요한 교육을 제대로 받지 못했다고 생각한다.
- 3) IT 학문 분야의 정체성 결여로 해당분야 전문인력의 경쟁력이 약화되고 있다. 일례로, 프로그래밍은 누구나 쉽게 할 수 있다고 생각하는 반면, CST 전공자는 프로그래밍 밖에 할줄 모른다고 간주되고 있다.

현재 중국에서 IT분야는 더 이상 선호하는 직업이 아니며, 따라서 대학에서도 과거 중국학생들의 선호도

1순위였던 CST 전공이 현재는 5~6위로 추락하였다. 위원회는 이와 같은 현상의 원인으로 장비 및 시설의 부족, 교수의 부족, 진부한 커리큘럼 등의 세가지를 지적하였다. 특히 1991년부터 채택되어 사용되어온 커리큘럼(CC1991)으로는 날로 다양화 되고 있는 IT 전공의 모든 분야를 적절히 교육할 수 없으며, 30만명 내외의 학생들을 효과적으로 교육할 수 없다고 지적하였다.

중국에서도 각종 통계자료는 현재 뿐 아니라 향후에도 IT 전문인력의 공급 부족이 심각한 것으로 지적하고 있어, 중국 정부로서는 IT 분야 학생 공급의 감소는 시급히 해결해야 할 문제중 하나이다. 중국 정부 주도로 진행되고 있는 IT분야 학생 지원감소에 대한 대책은 모든 대학으로 하여금 IEEE-CS/ACM의 CC2001 커리큘럼을 채택하도록 하고, CST 전공을 전산학(Computer Science), 컴퓨터공학(Computer Engineering), 소프트웨어 공학(Software Engineering), 정보 기술(Information Technology)의 네 가지 전공으로 세분화하여 이들 중 하나로 특성화하도록 유도하고 있다. 특이할 만한 사항은 CST 전공의 70% 이상이 네 가지 세부 전공 중 정보기술 분야를 선호하고 있다는 점이다. 과거 Science나 Engineering 위주의 IT 전공이 실용적인 Technology를 추구하고 있음을 시사하고 있다.

4. 대학에서의 교과과정 운영의 두 가지 사례

이 장에서는 두 가지 사례를 제시함으로써, 국내 교육의 단면을 보일 뿐 아니라 상호 비교할 수 있는 기회를 제공하고자 한다.

4.1 전북대학교 컴퓨터공학 전공[8]

전북대학교 전자정보공학부는 1학년 2학기가 끝나는 시점에 컴퓨터공학, 전기공학, 전자공학의 세 가지 전공 중에 하나로 배정한다. 컴퓨터 공학 전공의 학생 정원은 130명이다. 교수는 20명이고, 교수의 전공 분야는 비교적 골고루 분포되어 있다. 공학인증(ABEEK)에 대비하여 2006년에 교과과정을 개편하였다. 공학인증은 2007년 하반기에 신청할 예정이고, 현재 시범 운영하고 있다. 개편 교과과정의 큰 특징은 소프트웨어 개발 과목을 강화한 점과 설계 과목을 운영한다는 점이다.

아래에서 프로그래밍 과목군과 소프트웨어 개발 과목군에 대해 학기와 과목이름을 나열하고 있다.

- 프로그래밍 과목군
- 1/2학기 C언어 기초
- 2/1학기 C++프로그래밍

2/1학기 Java 프로그래밍

- 소프트웨어 개발 과목군(*표시는 공학인증 설계과목)
- *2/2학기 시스템분석 및 설계
- 3/1학기 소프트웨어 공학
- *3/2학기 소프트웨어 개발 프로젝트 1
- *4/1학기 소프트웨어 개발 프로젝트 2

프로그래밍 기술을 2학년 1학기까지 충분히 익히도록 배치하였다. 자료구조와 알고리즘 과목은 각각 2학년 2학기과 3학년 1학기에 배치하여 프로그래밍을 숙지한 후 배울 수 있도록 배려했다. 이 두 과목에서는 트리, 그래프, 연결 리스트 등을 사용하고, 충분히 큰 데이터 집합을 입력으로 사용하는 프로그래밍 과제를 여럿 부과함으로써 프로그래밍 실력을 한 단계 높이는 기회로 삼게 하고 있다. 또한 주어진 문제와 응용 상황에 적합한 자료구조와 알고리즘을 설계하고 구현하는 능력을 충분히 배양하도록 유도하고 있다. 이러한 훈련 이후에 소프트웨어 개발 과목을 수강하게 함으로써, 효율적인 알고리즘 설계 구현 능력을 갖춘 상태에서 팀워크에 의한 소프트웨어 개발 과정에 진입하도록 유도하고 있다.

시스템분석 및 설계 과목에서는 소프트웨어 설계에 대한 개념을 습득하고, UML과 같은 소프트웨어 개발 툴을 사용하여 소프트웨어 제품을 설계 제작하도록 한다. 국내의 소프트웨어 공학 과목이 추상적인 수준의 교육에 머무는 경우가 적지 않다고 알려져 있다. 소프트웨어 공학을 배우기 직전 학기의 시스템분석 및 설계 과목에서 도구 사용에 의한 설계 제작 훈련은 이러한 장애를 충분히 극복해 줄 것이다.

소프트웨어 개발 프로젝트를 두 학기에 연속으로 배치함으로써 충분히 큰 소프트웨어 개발을 시도할 수 있도록 배려했다. 이 과목에서 개발한 설계도, 소프트웨어, 테스트 결과, 문서 등은 학생이 취업에 활용할 포트폴리오를 충실하게 해 줄 것이다.

또한 아래와 같이 과목 연계 체계를 갖추어 이론과 프로그래밍 연습이 이어지도록 배려했다. 프로그래밍 과목은 모두 공학인증의 설계과목으로 지정하여 단순 프로그래밍을 넘어 체계적인 소프트웨어 개발 훈련을 할 수 있도록 하였다(*표시는 공학인증 설계 과목).

- 3/1 운영체제 -> 3/2 *유닉스시스템 프로그래밍
- 3/2 데이터베이스 -> 4/1 *데이터베이스 프로그래밍
- 3/2 컴퓨터네트워크 -> 4/1 *네트워크프로그래밍
- > 4/2 *모바일 프로그래밍

4.2 전남대학교 컴퓨터공학 전공[9]

전남대학교는 2002년도에 전산학과, 컴퓨터공학과, 전자공학과 등 3개 학과를 통합하여 전자컴퓨터공학부를 신설하였으며, 컴퓨터공학 및 전자공학 두 개의 전공을 운영하고 있다. 전공배정은 2학년 진입시에 결정되며 컴퓨터공학전공에 20명, 전자공학전공에 11명의 교수가 있다. 컴퓨터공학전공에서는 2006년도에 공학교육인증(ABEEK)에 대비하여 교과과정을 대폭 개편하였다.

당 전공에서 과거 수년간 관측한 바에 의하면, 졸업이 압박한 학생들의 프로그래밍 능력이 절대적으로 낮으며, 이로 인해 학생들 스스로 전공분야 자신감 결여 및 산업체로의 취업 기피 등의 현상으로 이어지고 있다. 당 전공 소속교수들의 수차례 워크샵에 의해 도출된 대책은 학생들이 전공에 대한 자신감을 갖도록 해주는 것이며, 이는 전공교육의 강화로만 이뤄질 수 있다는 것이다. 이에 따라 2006학년도 교과과정 개편시에 아래와 같은 세 가지 프로그래밍교과목을 정비하고, 이들 중 C와 C++는 전공필수 교과목으로 지정하여 모든 학생이 수강하도록 하였다.

1/2학기 C프로그래밍기초및실습

2/1학기 C++프로그래밍및설계

2/2학기 자바프로그래밍및설계

프로그래밍교육의 내실을 다지기 위해 각 과목별로 대표교수를 선임하고, 모든 분반의 강의내용, 실습 및 성적 평가를 공동으로 운영하도록 하였다. 아울러 일정 기준에 미달하는 학생들은 그 비율에 관계없이 과락(F)시킴으로써 과거와 같이 부실한 프로그래밍 교육이 누적된 채로 상급 학년으로 진급되는 현상을 차단하고자 하였다.

일례로, 2006년도 2학기에 개설된 C프로그래밍기초 및실습의 경우 모두 7개 분반에 192명의 학생이 수강하여 46명(24%)의 학생이 F학점을 받았다. C프로그래밍기초및실습 교과목의 학습목표는 다음과 같으며, 이 중 네 번째 학습목표를 기준으로 학기말까지 적당한 난이도를 갖는 10라인 정도의 C프로그램을 작성하는 수준에 이르지 못하는 경우 F학점을 부여하는 것을 모든 분반의 평가 원칙으로 설정하였다.

- 1) 프로그램이 컴퓨터에서 실행되는 단계를 이해하고 자신의 프로그램을 실행시킬 수 있다.
- 2) 키워드, 변수, 자료유형 등의 기초 프로그래밍 용어를 알고 있다.
- 3) C언어의 연산자, 조건문, 함수, 배열, 문자열 처리 등을 이해하고 응용할 수 있으며, 구조체, 포

인터, 파일처리 등의 기본 원리를 이해한다.

- 4) 50라인 정도의 프로그램을 설계하여 C언어로 작성할 수 있고, 다른 사람이 작성한 100라인 정도의 C 프로그램을 이해하고 출력을 예측할 수 있다.

5. 노력할 점

3장에서 학생들이 프로그래밍을 기피하는 현상과 그 이유에 대해 살펴보았다. 여기서는 이러한 문제를 해결하기 위해 취할 수 있는 또는 반드시 취해야 하는 몇 가지 방안을 제시한다.

5.1 초중고 컴퓨터 교육 내용의 개선과 강화

미래 지능 로봇, 인지 능력을 갖춘 컴퓨터, 인간 친화형 인터페이스 개발을 위해서는 컴퓨터를 단지 공학으로만 보아서는 안 되고 과학 영역으로도 보아야 한다. 컴퓨터 과학은 학생들에게 문제 해결력, 창의력, 의사 결정력 등을 배양하는데 매우 효과적이다[10]. 이제는 초중고에서 컴퓨터 과정을 물리, 화학, 생물 정도의 중요성을 부여하여 가르쳐야 한다. 어려서부터 컴퓨터를 매우 흥미로운 학문으로 여기도록 하고 지적 호기심을 자극할 요소를 교육 과정에 포함시켜야만 한다.

이러한 측면에서 볼 때, 2007년 6월 25-27일 무주에서 열리는 한국정보과학회 KCC2007의 “초중고 컴퓨터교육 체계 개선안” 패널 토의는 매우 시의 적절한 대응이라고 생각한다.

5.2 ACM-ICPC 대회의 역할 확대

이 대회는 대학생들이 프로그래밍에 대한 적극성을 가지도록 자극하는데 매우 효과적이다. 현재는 소수의 뛰어난 학생들의 무대로 여겨지고 있는데, 횡수를 확대한다면 등급을 여럿 두어 참여 폭을 넓힌다면 정도의 노력을 통하여 국내 컴퓨터 학과 학생들에게 인지도를 높이고 진입 장벽도 낮출 필요가 있다. 최근 신설된 문제뱅크와 인터넷을 통해 맞았는지 여부를 자동으로 채점해주는 시스템 운영은 매우 바람직해 보인다.

5.3 교과과정 개편에 대한 유혹을 떨칠 것

아직도 교과과정 개편을 통해 문제를 해결하려는 사람들이 있다. 하지만 정보통신부의 교과과정 개편 사업, 교육부의 BK21, 공학인증, 그리고 최근 정보통신부의 NEXT 사업 등의 강요에 의해 교과과정을 수차례 바꾸어 온 것이 사실이다. 심지어 어느 순간에 어느 대학의 교과과정을 보면 네 개 학년의 것이 모두 다른 경우도 있다. 이 정도면 충분하다고 본다.

본질을 보자. 문제는 교과과정에 있는 것이 아니라 교과과정에 포진해 있는 과목을 제대로 운영하느냐에 있다. 고전적인 교과과정을 가지고도 얼마든지 현대적인 프로그래머를 양성할 수 있다. 예를 들어, 운영 체제라는 과목에서 고전적인 이론뿐 아니라 Linux 커널을 이용한 임베디드 소프트웨어 제작이라는 실습을 운영할 수 있다. 알고리즘 과목에서는 수십만 건의 웹 문서를 데이터로 주고 효율적인 검색 프로그램 제작을 과제로 부여하여, 현장의 정보 검색 응용에서 효율적인 알고리즘의 중요성을 느끼게 할 수 있다.

교수의 강의 능력의 제고도 중요한 요소이다. 교수가 고전적인 이론과 고전적인 프로그래밍 스타일만 갖추고 그에 따라 강의하는 것을 고집하는 상황에서는 교과과정을 개편해봐야 학생들에게 혼란만 주게 된다.

IEEE Computer Society와 ACM이 공동 개발하고 있는 컴퓨팅 커리큘럼을 보면, 우선 학생들에게 가르쳐야 할 지식 덩어리(knowledge body)와 이를 습득하는 데 필요한 시간을 같이 나열한 후, 이들을 교과목에 배치하는 방식을 사용하고 있다[11,12]. 내용을 먼저 확립하고 틀을 만드는 것이다. 최근에는 전 세계 대학 교수를 대상으로 의견을 받는 작업을 하고 있다. 계속 틀만 수정하고 있는 우리에게 시사하는 바가 많다고 생각한다.

5.4 저학년 프로그래밍 교육의 내실화

한국정보과학회에서는 학부기초 프로그래밍 교육 위원회를 2007년 초에 신설하여 연구를 진행하고 있다. 이 위원회는 KCC2007에서 “대학 저년차 프로그램 교육 개선안”이라는 패널 토론을 주도하였다. 이러한 움직임은 늦은 감이 없지 않지만, 매우 고무적이다.

이 위원회가 가장 고민할 점은 프로그래밍이 어렵고 재미없는 과목이 아니고 쉽고 재미있는 작업이라는 인식을 어떻게 심어줄 것인지에 있다. 이 측면에서 중요하게 고려할 사항은 우리나라 대학 신입생의 상태와 입장을 충분히 이해하는 것이다. 고교 3년을 통해 파김치 상태가 되어 대학에 들어 온 점, 그리고 단순 암기 위주의 교육을 받아 왔기 때문에 무엇인가를 스스로 만든다는 것에 대한 익숙지 않음과 두려움을 가지고 있다는 점이다. 대부분 학생이 입출력, 연산, 제어 흐름, 함수 각각은 잘 다룬다. 하지만 이들을 서로 짜 맞추어 주어진 문제를 해결하는 것은 매우 어렵다. 이러한 학생들에게 흥미를 유발시킬 수 있는 방식의 교재 개발도 매우 중요한 작업으로 보인다.

5.5 기업 인턴십 강화

현재처럼 개발 환경이 급변하는 상황에서는, 대학

에서 기업이 요구하는 수준의 즉시 활용 가능한 소프트웨어 개발자를 키우는 것은 쉬워 보이지 않는다. 따라서 기업 인턴십은 필수적이다.

대부분 정부 사업이 인턴십을 요구 또는 강제화하고 있다. 옳은 방향이다. 하지만 현실을 보면, 형식적인 흉내로 실적을 채우는 경우가 적지 않은 것이 사실이다. 이러한 상황을 극복하지 못한다면 거의 모든 논의가 물거품이 될 가능성이 높다.

2006년 ACM-ICPC 대회장에서 NHN이 대회에 출전한 학생에게는 인턴십에 지원할 자격을 주고 있는 장면을 목격하였다. 기업이 인턴십 자체를 꺼리는 것이 아니라, 자격 미달인 학생을 받는 것을 극도로 기피하고 있는 것이다. 따라서 기업과 대학이 머리를 맞대고 인턴십 채용 규격을 만들 필요가 있어 보인다. 대학은 그 규격을 만족하는 학생만을 선발하여 인턴십을 보내고, 학생들에게는 한 학기 인턴십을 졸업 요건으로 강제화하여 학생들이 그 규격을 만족하지 못하면 졸업할 수 없도록 제도화하는 것이다.

5.6 프로그래머라는 직업의 사회적 인식 제고

이 문제는 현실적으로 가장 중요하다고 생각한다. “한전” 때문에 전기 전공에 지원하고 “삼성 전자” 때문에 전자 전공을 지원한다는 학생을 보면 미래에 대한 그들의 짧은 안목에 대해서도 실망스럽지만 프로그래머라는 직업이 사회적으로 낮게 인식되는 점에 대한 책임감도 느끼게 된다. 이 문제는 대학 혼자서 절대 해결할 수 없고, 정부 그리고 기업과 머리를 맞대고 해결책을 모색해야만 한다.

현재 프로그래밍은 진입 장벽이 매우 낮아 보인다. 6~12개월 동안 학원에서 프로그래밍을 훈련하고도 4년제 대학 졸업생과 같은 대우를 받는 경우가 허다하다. 정보처리기사는 몇 개월 집중적으로 공부하면 취득할 수 있는 자격증이 되어 있다. 한마디로 컴퓨터 프로그래밍이라는 분야가 가진 고유의 전문성이 우리 사회에서 통하지 않고 있는 것이다. 당연히 소프트웨어 기업들은 질 낮은 소프트웨어를 3~4개월 만에 개발 납품하여 근근이 연명하는 수준에 머물러 있다. 이런 과정에서 프로그래머는 밤새우며 고생하지만 급여는 불만족스러운 직업으로 전락해 버렸다. 이러한 상황을 타개하는 하나의 방법은 정부가 엄격한 규격에 따른 프로그래밍 전문가 자격증제도를 만들어 프로그래밍에 대한 품질 관리 체계를 만드는 것이다.

또한 우리나라 사람들의 무형물에 대한 가치 인정이 인색한 문화가 소프트웨어는 값싼 것 또는 공짜로 얻을 수 있는 것이라는 인식을 팽배하게 만들었다. 최

근 정부부터 나서 소프트웨어 제값 주가를 실천하는 것은 고무적인 변화로 보여 진다.

6. 결론

매년 김구 기념관에서 열리는 ACM-ICPC 대회에 가 보면, 학생들의 살아있는 열정과 자부심이 피부로 느껴진다[13]. 문제 풀이에서는 단기 학원 공부 또는 전자와 같은 다른 전공 공부로 쉽게 흉내 낼 수 없는 창의성이 요구됨을 알 수 있다. 컴퓨터 학문 자체가 정체성을 상실하고 있는 것이 아니라, 학문의 변화 추세를 읽지 못하는 낡은 교육 방식이 문제를 일으키고 있는 것이다.

참고문헌

- [1] NSF, National Software Forum 2006, 르네상스 서울호텔, 2006년 7월 12일
- [2] 한국정보과학회 학회지(특집: 산업을 변화시키는 소프트웨어 기술), 2007년 2월
- [3] 한국정보과학회 학회지(특집: 임베디드 소프트웨어 기반 기술), 2006년 8월
- [4] 허철호, 컴퓨터 과학 중심의 중등학교 컴퓨터 교과과정, 전북대학교 교육대학원 석사 논문, 2006년 8월
- [5] 한국정보과학회 학회지(특집: 모바일 플랫폼), 2006년 7월
- [6] 한국정보과학회 학회지(특집: IT 인력 양성 정책), 2004년 5월
- [7] X.-M. Li, "Undergraduate computing education in China: a brief status and perspective," Journal of Harbin Institute of Technology, Vol. 12, pp. 4-5, 2005
- [8] 전북대학교 전자정보공학부 컴퓨터전공 2006 교과과정, <http://ei.chonbuk.ac.kr/>
- [9] 트랙별심화과정 개발 3차년도 결과보고서, 전남대학교 전자정보가전인력양성 누리사업단, 2007년 2월
- [10] 이미숙, 김갑수, "초등학교 컴퓨터 독립 교과 적용을 위한 현 컴퓨터 교육 과정의 문제점 및 개선 방안 고찰," 한국정보교육학회 학술발표논문집, 10권 2호, pp. 41-50, 2005년 8월

- [11] 오일석, 박혁로, "연구-교육 인력 양성을 위한 소프트웨어 교과과정," 한국정보과학회 학회지, 2001년 12월.
- [12] IEEE-ACM Joint Task Force, Computing Curricula 2001, (http://campus.acm.org/public/comments/comments_cc2001.cfm)
- [13] ACM-ICPC 대회 공식 사이트, <http://acm.kaist.ac.kr/forum/>



오일석

1984 서울대학교 컴퓨터공학과(학사)
 1991 한국과학기술원 전산학과(박사)
 1991~현재 전북대학교 전자정보공학부(교수)
 1996~1997 캐나다 Concordia 대학 CENPARMI 연구소(방문교수)
 관심분야: 패턴인식, 컴퓨터비전
 E-mail: isoh@chonbuk.ac.kr



김수형

1986 서울대학교 컴퓨터공학과(학사)
 1988 한국과학기술원 전산학과(석사)
 1993 한국과학기술원 전산학과(박사)
 1990~1996 삼성전자 멀티미디어 연구소(선임연구원)
 2000~2002 캐나다 Concordia대학 CENPARMI 연구소(방문교수)
 1997~현재 전남대학교 전자컴퓨터공학부(교수)
 관심분야: 인공지능, 패턴인식, 문서영상검색, 유비쿼터스컴퓨팅
 E-mail: shkim@chonnam.ac.kr



양재동

1983 서울대학교 컴퓨터공학과(학사)
 1985 한국과학기술원 전산학과(석사)
 1991 한국과학기술원 전산학과(박사)
 1995~1996 University of Florida, Visiting Scholar
 현재 전북대학교 전자정보공학부 교수
 관심분야: OODBs, Expert System, CASE, 온톨로지
 E-mail: jdyang@chonbuk.ac.kr