

電算
教育

教科課程의 問題点과 改善方案



元 裕 憲
〈弘益大 電子計算學科 敎수〉

◇ 대학 교과과정

우리나라 전자계산학과와 교과 목록은 개설된 과목수가 미국에 비하여 많기 때문에 ACM의 전자계산학 교과목 위원회(Curriculum Committee on Computer Science of the ACM, 이하 C'S라 칭한다)에서 추천한 것에 비하여 소수의 선택 과목만이 개설되지 않았을 뿐이다. 우리나라에서 강의가 별로 개설되지 않은 과목은 알고리즘, Artificial Intelligence, Automata and Formal언어 등 최근에 발생된 과목이나 매우 이론적 과목에 국한 되어 있다. 이런 과목들은 이론에 치우친 과목을 제외하고는 곧 대학 교과 과목에 다 들어오리라 생각된다.

더구나 1979년 1월에 중앙대학에서 전국 전자계산학과 교수들이 전자계산학과와 교과과정에 대한 심포지움을 문교부 지원하에 가졌었기 때문에 교과목들의 이름은 좀 다를지라도 C'S에서 추천한 과목들이 여러 과목으로 분리되어서 개설되어 있다. 그러므로 전자계산학과들의 문제점은 교과목의 누락에 있다고는 생각되지 않고 강의 내용에 있으며, 이는 신흥 학문으로서 어디에서나 일어나는 현상이다. 단지 그 당시에도 논란대상이 되었던 정보처리 1급 기사 과목들이 문제를 더욱 악화시켰다. 대학에서는 기사 시험을 무시할 수는 없으며 기사 과목은 전자계산학과에서 중시 되어야 할 프로그래밍 언어와 시스템 프로그래밍과 운영체제 과목들이 소

홀히 취급된 대신, 수치해석, 통계, OR과목들이 큰 비중을 차지해 왔다. 다행히도 금년 여름에 한국정보 과학회에서 정식으로 기사과목 변경 신청을 접수 시켰기 때문에 그 문제점은 내년부터는 해결되리라 생각된다.

참고로 변경 신청과목은 다음과 같다.

● 프로그래밍 언어 ● 자료구조 ● 시스템 프로그래밍 ● 전자계산기 구성 ● 통계 및 수치해석
상기 과목중에서 내용면에서 가장 큰 오류를 범해온 과목이 프로그래밍 언어에 대한 관념들이다. 전에는 FORTRAN중심에 COBOL과 어셈블리 문제가 섞인 정도로 취급되어 왔었다. 그러나 지난 7월16일에 충주 수안보에서 전자계산학과 교수회의가 있었으며, 그 자리에서 새로 변경되는 기사 과목들에 대한 내용 결정을 한 바 있다. 프로그래밍 언어 그룹에서 결정된 바로는 그 과목의 내용을 다음과 같이 하기로 하였다.

① 프로그램 언어의 구조에 관한 기초 사항 ② 프로그래밍 능력 평가 (PASCAL, FORTRAN COBOL) ③ 컴파일러 및 Formal언어 기초사항 ④ 스트럭처드 개념 등 프로그램 형태에 영향을 주는 Software Engineer 기본사항.

상기 서술된 내용은 실질적으로 전산학과 교육에서 프로그래밍 언어 계통으로 꼭 알아야 될 사항들을 망라 하였다. 이와 같은 현상은 자료구조에서도 반영되어 일부 알고리즘과 데이터베이스 과목내용을 많이 포함시키기로 결정하였

다. 그러므로 대학 전자 계산학과 과목의 내용 면에서도 점점 나아지고 있다.

◇대학교육의 기초과정

상기 추세로 보아서 전산학과의 교과과정의 문제점은 점점 사라지고 있으나, 아직도 가장 큰 문제점들은 전산학과의 필수 기초과목들의 내용에 있다.

C³S에서 추천한 필수 기초과목은 다음 8 과목이다. 여기서 과목다음(2-2-3)는 주당 강의 2시간, 실습 2시간으로 3학점 과목이라는 표시이다.

- CS 1. Computer Programming I(2-2-3)
- CS 2. Computer Programming II(2-2-3)
- CS 3. Introduction to Computer Systems (2-2-3)
- CS 4. Introduction to Organization(3-0-3) or (2-2-3)
- CS 5. Introduction to File Processing (3-0-3)
- CS 6. Operating Systems and Computer Architecture I(2-2-3)
- CS 7. Data Structures and Algorithm Analysis(3-0-3)
- CS 8. Organization of Programming Languages(3-0-3)

위 과목들은 우리나라에서도 여러 이름으로 개설되는 중요과목이다. 이는 단지 우리가 3시간 강의 3학점인데 반하여, C³S에서는 2시간 강의 2시간실습이라는데 차이가 있다. 상기 열거된 과목들 중에서 실습이 필요한 과목들에 이론 강의만 한다는 것은 학생들의 적응력을 크게 저하시킨다.

우리나라에서도 컴퓨터 프로그래밍은 어떤 형태로든 2-3 강좌를 행한다. 그럼에도 불구하고 대다수 학생들은 합리적인 시간안에 적당한 프로그래밍을 할 수 있는 학생들은 극히 드물다. 이 상황은 매우 심각하게 고려되어야 할 것이다. 저급학년에서 프로그래밍을 제대로 할 줄 모르

면 상급학년에 가서는 점점 더 전산학과에서 흥미를 잃게 되며, 좋은 알고리즘을 배운다 하여도 표현력들이 약해져 산 지식이 되지 못하고 곧 잊어 버리며 졸업 후에는 실제업무 적응력이 매우 약하여 전산학과 출신이나 타과 출신이나 똑 같다는 평을 듣게 되는 원인이 된다.

◇컴퓨터 프로그래밍 과목들의 문제점

단지 전산분야 뿐이 아닌 모든 분야에서 우리나라 대학생들이 학문의 실제응용에 매우 약하다고 한다. 이는 우리나라 교육이 어릴 적부터 암기위주 주입식교육에 큰 책임이 있겠으나, 더욱 중요한 것은 대학 저급 학년때 실습을 제대로 할 수 없다는데 기인된다고 생각된다. 각 대학의 기자재 부족과 실습을 도와 줄 조교의 부족과 자질이 전산학 쪽에서는 심각하다. 프로그래밍을 신청한 학생들은 컴퓨터 사용이 쉬워야 되는데 사용하기에 불편하면 집집마다 프로그램을 작성하여 시행시키지 않게 된다. 대다수의 학교에서는 Batch 시스템에 의존하기 때문에 천공이 문제가 되고, 한번 시행시키기 위한 시간이 오래 걸리어 오류가 몇번 발생되면 심한 경우 한달만에 결과를 받게 된다. 일부 학교에서는 터미널을 사용할 수 있으나 수강 학생수에 비하여 터미널 수가 너무 적어서 10분을 쓰려면 2~3시간 기다리는 실정이 대부분이다. 상황이 이렇게 되니 강의를 맡으신 분들은 실습을 시키는데 제한을 받을 뿐 아니라 학생들 자신은 의욕 상실되어 버린다.

C³S에서 프로그래밍을 수강한 학생들에게 요구하는 사항은 다음과 같다.

합리적인 시간내에 읽기 쉽고 문서화가 잘된 적절한 프로그램 작성능력을 가져야 한다. 그들이 자신이 작성한 프로그램이 합리적인 효율성을 가졌는지 또 적절한 프로그래밍지를 평가 할줄 알아야 한다. 개인으로서 또는 팀 멤버로서 수행해야 할 일에 포함된 것들을 평가 할줄 알아야 한다. 상기 요구 사항에 대하여 우리 학생들은 너무나 거리가 먼 현실이다. 학생들이 걱정의 프로그램을 작성하지 못하는 것도 문제이지만 읽

기 쉬운 프로그래밍과 문서화를 하지 못하는 것은 더 큰 문제이며 이는 강의하는 분들이 노력하면 좋은 성과를 보리라 믿는다. 더구나 학생들이 자신이 작성한 프로그램을 평가하는 능력은 더욱 요원하다.

왜 이러한 상황이 발생하는지 C³S에서 제시한 과목 내용을 분석하여 우리와 비교하여 보자. C³S에서 필수기초전공과목에서 요구하는 것을 크게 대별하면 프로그래밍에 관한 사항, 소프트웨어 시스템에 관한 사항, 하드웨어 시스템에 관한 사항, 자료구조와 화일처리에 관한 사항으로 분류되며, 이 중에서 프로그래밍에 관한 사항과 화일처리에 관하여 요구하는 사항이 우리나라 현실교육과 큰 차이를 보이고 있다.

프로그래밍에서 요구하는 사항을 요약하면 다음과 같다.

P 1. 알고리즘 : 알고리즘의 개념과 성질, 문제에서의 알고리즘 역할, 알고리즘 표현 언어와 구성.

P 2. 프로그래밍 언어들 : 문제 중심의 고급어 (SNOBOL, APL, LISP 등등)의 기본 구분 및 개념, 부프로그램, 입출력문, Recursion.

P 3. 프로그래밍 형태 : 스트럭처드 프로그래밍의 개념 및 기술, 프로그램의 문서화

P 4. 디버깅과 확인 : 디버깅 소프트웨어 사용 능력, 테스트 데이터 선정 능력, 오류를 찾아내는 기술, 프로그래밍 확인.

P 5. 응용 : 정보검색과 화일 유지에 관련된 일부분제, Lexical Analysis, 문자처리와 수치계산, 프로그래밍 언어의 특징을 요구하는 문제.

국내의 컴퓨터 프로그래밍 과목에서는 FORTRAN과 COBOL 프로그램 작성과 스트럭처드 언어로 PASCAL을 배우기는 하나 대다수 실습을 못하고, 학교에 따라서 PL/I을 강의 하는 곳도 있다.

상기 P 1 - P 5에서 요구하는 사항들을 만족시키면, 기업에서 "전산학과 출신보다 타과 출신이 프로그래밍을 잘 한다."는 듣기 거북한 소리는 듣지 않을 것이다. 여하튼 문제는 강의하는 교수들 자신이 단지 프로그램 작성만이 아닌

문서화, 스트럭처드 개념, 테스트 방법, 알고리즘 등등을 더욱 다양하게 강의 해야 하며 실습 (우리나라 경우 숙제 대치)을 많이 시켜야 될 것이다.

◇ 프로그래밍 과목내용과 개선책

프로그래밍에서 첫 강좌는 고급어를 하나 선정하여 강좌의 반은 그 언어의 구문과 입출력문장들을 강의 하고, 나머지 반강좌는 스트럭처드 프로그래밍 기법에 의하여 간단한 알고리즘을 (수치계산 문제, 약간의 SORT와 Merge 문제, 자료를 다루는 간단한 상입문제)을 표현시키고 Flow Chart를 만들어 프로그래밍을 작성하여 실습시켜야 된다. 만일 FORTRAN과 같은 스트럭처드 언어가 아닌 언어를 가르칠 경우는 flow Chart를 스트럭처드 개념 (Nassischneiderman Chart 등)으로 하여 GOTO文의 사용을 제한하여 좋은 프로그래밍 습관을 갖도록 하여야 한다.

두번째 학기의 프로그래밍 과목에서는 큰 문제를 다룰 수 있도록, 프로그래밍 디자인, 디자인에 필요한 도구와 표현법 (Structured, Refinements, 간단한 HIPO 개념 등), 디버깅과 테스트에 집중 교육하며, 간단한 자료구조, SORT, SEARCH 등에 관한 알고리즘들을 분석 평가하며, 문자처리와 Recursion 개념도 소개한다.

상기 내용을 두 강좌에 어려우면 세 강좌를 개설하여서라도 가능한한 다 다루어 주어야 할 것이다.

필자가 FORTRAN 한 강좌는 수강한 학생들에게 프로그래밍 과목을 맡아 다음과 같이 강의 해본 결과 매우 바람직 하다고 생각 되었다. 학교에 스트럭처드 프로그래밍이 가능한 언어가 없어 구문이 간단한 PASCAL을 강의 하면서, 많은 알고리즘들을 제시하고 스트럭처드 개념, 하향식 프로그래밍, Refinements 방법으로 프로그래밍을 디자인하여, Nassischneiderman Chart로 프로우 차트를 작성케 한 뒤에 PASCAL로 프로그래밍하고서 FORTRAN으로 기계적인 변경을 하여 실습을 시켰다.