

논문 2017-1-8

# 소프트웨어 구성능력 향상을 위한 C-언어 교육방법 개선에 관한 연구

이현창\* 최근영\*\*

## A Study on the Improving Education Method of C-language for the Enhancing Ability of Constructing Software

Hyun-Chang Lee\* Keun-Young Choi\*\*

### 요 약

본 논문에서는 소프트웨어 구성능력 향상을 위한 개선된 교육방법을 제시하였다. 이를 위해 현재 시행되고 있는 소프트웨어 구성용 언어의 학습에서 존재하는 문제점을 분석하고 이에 대한 해결방안을 제시하였다. 제시한 방법의 목적은 C-언어 교육이지만, 여기에 HTML 결과화면과 웹-페이지의 흥미를 더하고, 모든 학습자가 다른 학습자들이 구성한 웹-페이지를 살펴볼 수 있어 학습에 대한 흥미가 더 증대될 수 있다. 제시한 교육방법의 효과를 확인하기 위해 펜티엄-PC에 레드-햇 리눅스, 아파치 HTTP 데몬 등과 함께 웹-서버를 구성해 다년간 학습을 진행한 결과 학생들의 성적이 크게 상승하는 효과가 있음을 확인하였다.

### Abstract

In this paper, improved education method for the enhancing ability of constructing software was proposed. Although purpose of proposed method is C-language education, but it include the interesting display result of HTML and the fun of Web-page. On the other hand, as all of student can watch Web-pages of other student, then interest of education can be increased. In order to verify the effectiveness of proposed education method, experimental Web-server is constructed as Pentium-PC with Red-Hat Linux, Apache HTTP daemon and gcc compiler. Experimental results show that the record of student was increased highly.

**한글키워드 :** 소프트웨어 교육, 프로그래밍, C-언어, CGI, 웹-페이지, HTML

\* 공주대학교 정보통신공학부  
(email: hcllee@kongju.ac.kr)

\*\* 공주대학교 시간강사  
접수일자: 2017.06.02 수정완료: 2017.06.28

## 1. 서 론

소프트웨어를 구성하기 위해 사용되는 컴퓨터

언어는 그 용도에 따라 다양한 종류가 있고, 시대적 요구에 따라 학습 대상으로 주목받는 언어의 종류 또한 변화한다. 예를 들어 초창기 컴퓨터는 매우 고가였으므로 학교나 기관, 기업체 등 특수한 계층에서 특수한 용도에만 사용되었기 때문에, 이들 컴퓨터에 사용되는 소프트웨어를 작성하기 위한 언어는 주로 공학계산을 위한 포트란(FORTRAN)과 사무용 언어인 코볼(COBOL) 등이 주류를 이루었다. 이후 컴퓨터 기술의 발전에 힘입어 PC(Personal Computer)가 출현하고 대중화됨에 따라 일반인이 컴퓨터 언어를 쉽게 접할 수 있는 교육용 언어인 BASIC(Beginners All-purpose Symbolic Instruction Code)과 PASCAL 같은 언어가 주목을 받았다<sup>[1]</sup>. 현대에 와서는 컴퓨터를 이용해 사무 자동화를 비롯해 인터넷 통신은 물론 게임과 같은 소프트웨어의 제작에 적합한 C-언어에 관심이 집중 되었지만<sup>[2-5]</sup>, 이 언어는 본래 시스템 기술용으로 개발된 것으로서, 기본적으로 어셈블리 언어의 개념을 기반으로 하기 때문에 포인터(pointer)와 같은 어려운 부분에서 많은 피교육자들이 포기하는 요소가 된다. 최근에는 스마트-폰의 등장과 이를 운영하기 위한 안드로이드 기반의 운영체제가 대중화 되면서 C-언어에 비해 개념적으로 더 쉬운 Java 언어가 관심을 받고 있으며, Python 언어 또한 점차 관심이 커지고 있는 추세이다. 사실 이러한 컴퓨터 언어들은 비교적 오래 전에 개발되어 각 분야에서 나름대로 역할을 해 오던 것이지만, 시대적 요구에 따라 관심의 대상이 달라진 것뿐이며, 아직은 여전히 C-언어가 가장 핵심적인 언어로 자리 잡고 있으며, 이 때문에 공학계열 교육기관에서는 거의 필수적으로 C-언어에 대한 교육과정이 존재한다. 특히 C-언어는 시스템 기술용이 본래 목적이었으므로 기기의 하드웨어 등을 섬세하게 제어할 수 있는 것 때문에 산업현장이나 소프트웨어 개발업계 내에서 C-언어

의 중요성은 당분간은 변화하지 않을 것으로 보인다. 그러나 C-언어의 이러한 뛰어난 기능으로 인해 오히려 학습자 입장에서는 어려운 언어로 인식되고 있으며, 교육현장에서도 매년 수많은 학생들을 대상으로 교육이 이루어지고 있으나 정작 소프트웨어 제작에 대한 우수한 인재는 극히 드문 것이 현실이다.

본 논문에서는 소프트웨어를 구성하기 위해 매우 중요한 C-언어가 어떠한 요인으로 교육효과가 낮은지 그 원인을 분석하고, 이를 토대로 C-언어 교육효과를 개선하기 위한 새로운 학습방법을 제시하고자 한다.

## 2. 새로운 C-언어 학습방법의 제시

### 2.1 C-언어 학습의 장애요소 분석

#### 2.1.1 전문가와 초보자의 격차 문제

일반적으로 C-언어 교육자는 해당 분야의 전문가인데 비해 초보자는 처음으로 소프트웨어를 구성하는 언어를 접하는 상태이다. 특히 소프트웨어를 구성하는 언어는 해당 내용을 단순히 암기하는 것만으로 실력이 갖추어지는 것이 아니라 주어진 일을 분석하고 이를 해결하기 위한 방법을 도출해 내는 과정이 필요하며, 전문가의 입장에서는 이러한 능력이 이미 습득이 되어있는 상태이지만 이 능력은 전달할 수 있는 지식의 종류가 아니기 때문에 교육 초기에는 단순한 지식 전달만 이루어질 뿐이다. 따라서 지식적 관점에서 아닌 임무의 분석과 해결방법의 도출 능력 차원에서 교육자와 피교육자 사이의 매우 큰 격차로 인해 이를 극복하기 어렵다. 즉, 교육자가 학습자의 눈높이를 맞추기 어려운 면이 있다.

#### 2.1.2 C-언어는 어렵다는 사실의 인지

C-언어는 Assembly-언어를 제외한 다른 언어

에 비해 어려운 것이 사실이다. 예를 들어 BASIC 언어(Visual BASIC과 같은 변형된 것이 아닌 원형 BASIC)나 PASCAL 등과 비교하면 상당히 어렵다. 그러나 일반적으로 C-언어 교육서에는 C-언어의 특징은 "쉽다"라 강조되어 있지만, 소질이 없는 학습자에게 결코 쉽지 않은 언어이다. 교육서에서 강조하는 쉽다는 표현은 교육서의 저자가 학습자에게 "이는 쉽고 별 것 아닌 것이므로 도전해 보라"는 격려의 의도로 기록한 것이지만, 이러한 격려가 오히려 학습자에게 역으로 작용하는 것으로 보인다. 학습자가 학습 중 C-언어의 내용 중 포인트 같은 부분을 접했을 때 이는 본래부터 이해를 위한 상당한 노력이 필요한 부분이지만, 학습자 입장에서는 여러 교재들에서 C-언어는 쉽다고 표현하지만 자신은 어렵게 느껴지므로, 자신에게 문제가 있는 것으로 생각해 오히려 쉽게 포기하도록 만든다. 즉 남들은 다 쉽다는데 나만 어렵게 느껴지는 역효과가 나타난다. 처음부터 솔직하게 C-언어는 어렵다고 인식시켜 준다면 비록 처음부터 시작조차 하지 않고 포기하는 사람도 있겠지만, 학습자가 어려운 내용을 접하더라도 '원래부터 어려운 것'이라는 생각이 있다면 쉽게 포기하지 않고 이해하기 위해 더욱 노력했을 지도 모른다.

### 2.1.3 실행결과의 시각적 효과

학습자가 C-언어 학습에 어려움을 겪는 것 중 하나는 C-언어 프로그램의 실행 결과가 일반적으로 글자 위주로 진행되어 시각적 효과가 없으므로 학습자의 성취감이 그다지 크지 못하다는 것이다. 이에 따라 한번 작성한 프로그램을 더 이상 변경하거나 응용할 만한 요소도 발견되지 않으며 학습자의 욕구도 발생하지 못해 단순히 교습자의 지시에 의해 따라하는 형식이 될 수밖에 없다. 따라서 프로그래밍의 가장 중요한 요소인 학습자의 응용 능력이 배양되기 어렵고 이는

지속적인 학습 의욕 또한 저하시키는 결과를 초래한다. 이러한 결점들은 교재나 교습자의 문제에 있다는 것 보다는 C-언어의 특성 상 본질적으로 나타나는 문제이므로 단독으로 C-언어만을 학습하는 것은 학습자로 하여금 상당한 노력과 인내를 필요로 한다. 시각적 효과를 노리기 위해 종종 그래픽 기법을 동원하기도 하는데, 이는 표준 C-언어 등에는 없는 기능이므로 활용하기에 한계가 있고, 또한 C-언어의 고유 기능이 아니기 때문에 응용에 한계가 발생한다.

### 2.1.4 학습 목표의 부재

일반적인 C-언어 교습은 최종적인 목표를 제시하기 어렵고, 예를 들어 1부터 10까지 더하기, 구구단 프로그램, 급여 계산과 같은 예제를 사용할 수밖에 없으므로 학습자의 학습 의욕을 고취시키기 어렵고 학습 효과도 떨어진다. 그러나 이 또한 C-언어의 특성 상 필연적인 현상으로서 그래픽<sup>[6]</sup> 등 고급 기법을 동원한다 할지라도 간단한 도형 정도를 그리는 수준의 예제밖에는 되지 않으며, 실용 그래픽 예제를 구성한다 할지라도 그 구현이 어렵고 컴퓨터 기종에 지나치게 의존하므로 C-언어 고유의 특징인 이식성이나 호환성 면에서 바람직하지 않다. 즉 학습자가 무엇을 위해, 어떤 목표를 위해 C-언어를 공부해야 하는지 명확한 목표제시가 어렵기 때문에 학습자도 막연한 상태에서 학습을 진행해야 하며, 따라서 학습자가 어떤 소프트웨어를 만들고 보고자 하는 의욕이 없으므로 어떤 소프트웨어가 완성되더라도 이를 변경하거나 다른 곳에 응용하려는 의욕에 한계가 발생한다.

### 2.2 장애요소 극복방법의 제시

이상에 분석한 문제점들을 해결하기 위한 다른 학습법을 제시한다.

### 2.2.1 HTML과의 비교

인터넷 홈-페이지를 구성하기 위해 사용하는 HTML(Hyper-Text Mark-up Language)<sup>[7]</sup>은 그 자리에서 홈-페이지를 작성해 곧바로 결과를 확인할 수 있다. 즉 HTML은 인터프리터 언어이기 때문에 문서를 작성하던 웹-브라우저를 통해 도중 결과를 확인하는 것이 가능하다. 설령 작성한 문서의 뒷부분에 오류가 있다 할지라도 그 앞까지는 결과가 나타난다. 더구나 그 결과는 글자 크기, 색상, 표, 그림 등 다양한 수단을 이용하기 때문에 C-언어의 글자 위주 결과에 비교할 수 없을 만큼 흥미롭다. 따라서 학습자가 스스로 글자의 크기나 색상을 변경하고 표나 화면의 바탕색을 변경하는 등 응용의 학습욕구가 자연히 발생한다.

이에 비해 C-언어는 컴파일러 언어이기 때문에 소프트웨어의 기본 골격이 완성되지 않으면 프로그램 실행문제 이전에 컴파일조차 되지 않는다. 즉 소프트웨어의 전반부는 모두 정상일지라도 후반부에 한 개의 오류가 있다면 컴파일은 거부된다. 즉 소프트웨어의 작성과 수정과정이 지루하고 따분하며, 프로그램 작성이 완료된 후 컴파일 때 어떤 오류 메시지가 뜰 것인지 두려움부터 앞서는 것이 사실이다. 또한 어렵게 컴파일 과정을 넘겼다 하더라도 정작 소프트웨어를 실행하면 컴퓨터가 다운되거나 오동작이 발생했을 때 어느 부분에서 그러한 결과가 발생한 것인지 찾기 매우 어렵다.

### 2.2.2 HTML과 C-언어의 병행

만약 C-언어로부터 앞서 고찰한 HTML을 발생하고 이를 웹-브라우저에서 해석해 웹-페이지 화면으로 출력된다면 실행 결과에 대한 시각적 효과가 크게 좋아질 수 있다. 또한 학습자는 웹-페이지 화면을 변경하기 위해 C-언어를 변경해야 하기 때문에 학습자의 필요에 의해 자연스럽게

게 C-언어를 변경해야 하는 과정을 거치므로 변경과 응용의 동기가 충분히 부여될 수 있다.

실제로 C-언어 강의에서는 피교육자들이 강의가 끝나면 곧바로 교육장에서 퇴실을 하지만, HTML 강의에서는 강의가 끝났음에도 불구하고 자신의 웹-페이지를 더 수정하기 위해 교육장에 남아있는 경우가 많다. 이것이 학습동기의 유발이다.

### 2.2.3 HTML, C-언어와 유닉스

앞서 고찰한 C-언어를 이용한 HTML 발생은 일반적인 C-언어 컴파일러를 이용하면 충분히 가능하지만, 여기에 약간의 유닉스 지식<sup>[8]</sup>을 가미하면 가상적인 웹-페이지가 아닌 실제의 웹-페이지 제작이 가능하기 때문에 상당한 동기유발이 가능하다. 즉, 단순히 HTML 문서만을 만들어 이를 브라우저로 확인하는 것과, 학습자 자신이 작성한 HTML 문서를 실제 웹-서버에 올린 후 서버로부터 자신의 웹-페이지를 확인하는 것은 흥미 유발에 상당한 차이가 있으며, 특히 서버를 이용해 자신이 제작한 웹-페이지 이외에 다른 학습자들이 제작한 웹-페이지를 둘러보며 자신의 웹-페이지를 수정하는 것은 학습의욕을 유발하는데 상당한 효과가 있다.

### 2.2.4 CGI 프로그래밍

이상에서 살펴본 HTML, C-언어, 약간의 유닉스 지식이 총 망라된 CGI(Common Gateway Interface)<sup>[9]</sup>를 활용하면 학습자의 흥미를 유발하는 유용한 예제들을 제작할 수 있다. 예를 들어 학습자의 웹-페이지에 C-언어를 이용한 방문자 카운터를 제작해 장착하고, 몸무게와 신장을 입력하면 비만도를 판정해 주며, 다른 사람이 글을 남길 수 있는 방명록을 직접 제작해 운영하고, 간단한 설문조사 및 결과를 표시해 주며, 쇼핑 카트를 훑내 낼 수도 있다. 이러한 흥미로운 예

제를 제작함으로써 학습자가 C-언어 프로그래밍을 해야 하는 목표가 확실해지고 그 결과를 꼭 확인하고 싶으며 이를 변경하고자 하는 욕구가 발생한다. 이러한 과정을 지속하면 학습자 자신도 모르는 사이에 C-언어 프로그래밍 능력이 배양될 수 있다.

### 2.3 소프트웨어 교육의 진행방법

소프트웨어 교육을 진행하는데 있어 몇 가지 학습효과를 증진시키는 방안이 있으며, 이를 지키면서 학습하면 더 효과적일 수 있다.

첫째, 소프트웨어의 리스트는 피교육자가 직접 입력하도록 한다. 일부 교육서는 그 서적에서 사용하는 예제 프로그램 리스트를 CD나 인터넷 사이트를 통해 배포하는데, 프로그램 리스트를 입력하지 않고 눈으로만 보아서는 프로그램을 자세히 살필 수 없어 겉도는 학습밖에 되지 않는다. 즉, 하나하나 직접 입력하면서 프로그램의 구석 구석을 빠짐없이 살펴볼 수 있다. 또한 잘못 입력해 그 오류를 찾아내는 동안에 실력이 향상될 수 있다.

둘째, 한 테마가 정해지면 최종적으로 완성된 프로그램이 아닌 기본 골격으로부터 점차 개량해 나가며 진행한다. 예를 들어 웹-페이지에 설치하는 방문자 카운터 소프트웨어 제작을 목표로 하는 경우, 초기에는 오직 기본적인 방문자 카운터 기능만 제작하고, 이후 누적 방문자와 오늘 방문자를 구분하는 형식으로 확장하며, 최종적으로는 방문자의 IP-주소, 접속시간 등의 기록이 남도록 한다.

셋째, 기존 교재에서의 문법별 내용전개가 아닌 테마 중심으로 진행해 학습효과를 증대시킨다.

## 3. 적용결과 분석 및 고찰

제시한 소프트웨어 교육방법의 시행을 위해 Pentium-III와 Red-Hat Linux를 이용해 웹-서버 [10-12]를 구성하고 다년간 다수의 학생들을 대상으로 교육을 시행하였다. 그림은 교육과정 중 학습자가 구성한 웹-페이지 화면의 예로서, 동일한 구구단 프로그램이라 할지라도 발생하는 화면이 HTML로 구성되기 때문에 학습의욕과 동기가 크게 향상되었다. 그림 1은 실습용 웹-페이지의 첫 화면으로서, 모든 학습자들의 웹-페이지가 나열되어 있고, 자신의 웹-페이지뿐만 아니라 다른 학습자들의 페이지도 모두 확인할 수 있어 자신의 페이지를 더 잘 꾸미고자 하는 동기유발이 가능하다.



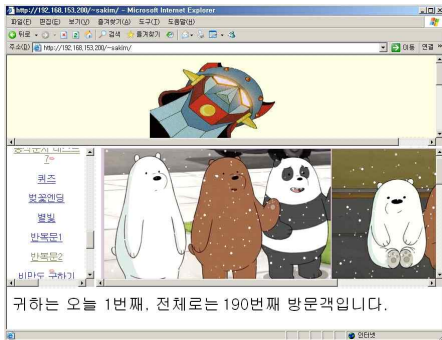
그림 1. 실습용 웹-페이지 첫 화면 예

그림 2는 학습자가 실습 중 작성한 웹-페이지의 예로서, 그림 2(a)와 2(b)에 나타난 바와 같이 각 학습자가 자신만의 독특한 그림이나 구성으로 구성할 수 있어 확실한 학습 동기유발 효과와 응용 욕구를 이끌어낼 수 있다. 특히 그림 2의 웹-페이지 하단 방문자 카운터 등은 모두 C-언어로 구성하였으며, 이를 구성하는 과정에서 프로그래밍의 목적과 성취감을 향상시킬 수 있다.

그림 3은 학습자가 실습하던 화면의 예로서, 그림 3(a)는 소프트웨어 실습에 항상 등장하는 구구단 프로그래밍의 결과이나, HTML과 결합됨으로 인해 다양한 색상과 화면 배치가 가능해 시각적 효과가 확실해짐을 알 수 있다.

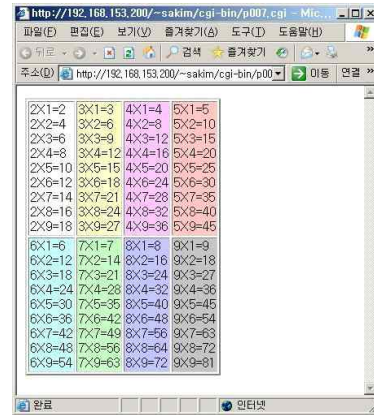


(a) 학습자-A의 웹-페이지 화면 예

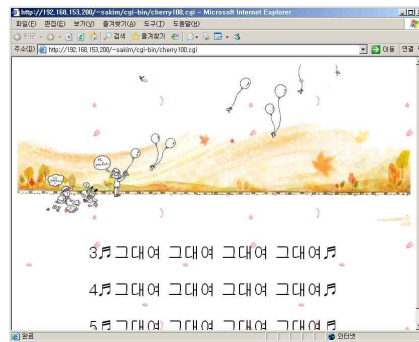


(b) 학습자-B의 웹-페이지 화면 예

그림 2. 학습자 웹-페이지의 표시 예



(a) 구구단 실행 화면



(b) 반복문 실습 화면

그림 3. 학습자 실습화면 예

그림 3(b)는 for()문을 이용한 반복문 실습화면의 예로서, 반복문이 실행될 때 웹-페이지 화면 하단에서 노래 가사가 반복적으로 스크롤 되며 나타난다.

이상과 같은 기법을 이용해 매년 학습자의 절대평가 점수 중 80점 이상에 해당하는 수를 표 1에 나타내었고, 이의 그래프를 그림4에 나타내었다. 여기서 (-)로 표기된 연차는 본 방법을 적용하기 이전을 의미한다.

표 1과 그림 4의 결과를 살펴보면, 비록 매년 시험문제의 난이도가 상이한 점을 감안하더라도 제시한 방법을 적용하기 전과 후의 성과 성적 우수 인원 비율은 약 410 % 증가하였고, 반 평균점수 또한 33 % 이상 증가해 뚜렷한 차이가 있음을 알 수 있다.

표 1. 적용 전과 후의 학생 성적표

년차	수강인원	80점 이상		반 평균
		인원 수	%	
-3	36	2	5.56	44.3
-2	30	2	6.67	45.7
-1	30	1	3.33	45.8
1	35	9	25.71	63.7
2	32	9	28.13	62.1
3	36	9	25	60.3
4	35	11	31.43	63.3
5	28	6	21.43	56.7
6	27	7	25.93	56.3
7	30	9	30	60
적용 전 평균	-	1.667	5.185	45.27
적용 후 평균	-	8.571	26.80	60.34
증가율[%]	-	414.29	416.92	33.31

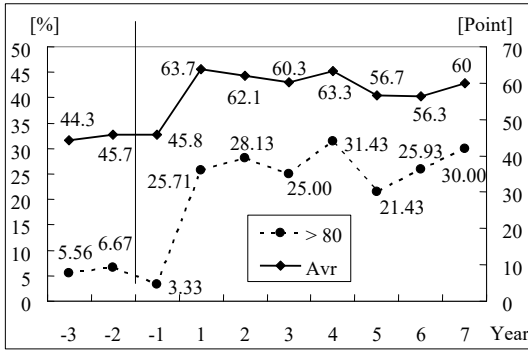


그림 4. 적용 전과 후의 성적변화 그래프

#### 4. 결론

본 논문에서는 소프트웨어 구성에 가장 많이 사용되는 프로그래밍 언어인 C-언어를 교육함에 있어 교육효과를 높일 수 있는 방법을 제시하였다. 이를 위해 기존의 C-언어 교육의 문제점을 분석한 결과 문법 위주의 진행순서로 인해 학습 내용이 지루해지고 목표가 뚜렷이 드러나지 않으며, 특히 소프트웨어 프로그래밍의 핵심인 개발 및 응용능력이 배양되기 어려움이 있다. 이를 극복하기 위해 HTML과 C-언어를 결합해 시각적 효과를 높이고, 여기에 약간의 유닉스 지식을 더해 학습자간 성과물을 확인하는 과정에서 성취감과 이를 더 개선하고자 하는 응용능력이 크게 향상될 수 있다. 제시한 학습법의 효과를 확인하기 위해 간단한 서버에 리눅스를 설치하고 이를 통해 학습자들이 자신의 웹-페이지를 제작토록 하였으며, 제작된 웹-페이지에 C-언어를 이용해 방문자 카운터, 비만도 측정 프로그램, 설문조사 프로그램, 방명록 등 일반적인 인터넷상에서 자주 접하는 소프트웨어 내용들을 직접 제작토록 한 결과 학습자의 학습효과가 크게 향상되었음을 다년간의 학습운영 결과를 통해 확인하였다. 향후 순수한 소프트웨어 뿐 만 아니라 하드웨어와 연

계된 지식을 필요로 하는 학습자들을 위해 하드웨어를 기반으로 한 소프트웨어 프로그래밍 언어를 학습할 수 있는 방안을 마련한다면 C-언어 프로그래밍 인재의 육성 노력이 더 넓은 영역으로 확산될 수 있으리라 사료된다.

#### 참고자료

- [1] Timothy N. Trainor, Diane Krasnewich; Computers! (3rd Edition); Mitchell McGraw-Hill, 1992.
- [2] Brian W, Kernighan, Dennis M and Ritchie; The C Programming Language; Prentice-Hall International, 1988.
- [3] Herbert Schildt; C: The Complete Reference(4th Edition); McGraw-Hill, 2000. 5.
- [4] Herbert Schildt; C++: The Complete Reference(4th Edition); McGraw-Hill, 2002. 12.
- [5] H. M. Deitel, P. J. Deitel; C++ How to Program; Prentice-Hall, 1994.
- [6] James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes; Computer Graphics - Principles and Practice (2nd Edition); Addison-Wesley, 1993.
- [7] 이승호; 파워 유저를 위한 홈페이지 작성법; 복두출판사, 1998. 3.
- [8] Graham Glass and King Ables; UNIX for programmers and Users (2nd Edition); Prentice-Hall, 1999.
- [9] 김명주, 엄성용; Web-Programming; 이안출판사, 1998. 5.
- [10] Kebi 캠퍼스; 웹 구축 - Linux vs Windows 2000; 이안디지털리, 2001. 7.
- [11] 유재우, 최종명, 최재영, 안보희; 웹 개발자를 위한 서블릿 JSP; 이안디지털리, 2000. 8.
- [12] Leland L. Beck; System Software - An Introduction to Systems Programming (3rd Edition); Addison-Wesley Longman, 1997.

저 자 소 개



이현창(Hyun-Chang Lee)

1986 단국대 전자공학과 학사  
1989 단국대 전자공학과 석사  
1996 단국대 전자공학과 박사  
1996~2004 국립 천안공업대  
학 정보통신과 부교수.

2005~현재 국립 공주대학교 공과대학  
정보통신공학부 교수.

<관심분야> 멀티미디어 회로, 전동기제어회  
로, 마이크로프로세서, 임베디드 소프트웨어



최근영(Keun-Young Choi)

1983 서강대 국어국문학과 졸업  
2000 단국대 정보통신과 석사  
2014 공주대 정보통신과 박사  
수료

2001~현재 국립 공주대학교 공  
과대학 시간강사

<관심분야> 인터넷 응용, 웹 프로그래밍, 소  
프트웨어 프로그래밍