

컴퓨터 프로그래밍 과제 관리 시스템

정찬호*, 김세기*, 김희철*
*한국의외국어대학교 컴퓨터및정보통신공학부
e-mail : radanova@gmail.com

A Management System for Computer Programming Assignments

Chan-Ho Jeong*, Se-Gi Kim*, Hee-Chul Kim*
*School of Computer Science and Information Communications Engineering
Hankuk University of Foreign Studies

요 약

프로그래밍 과제는 자료구조 및 알고리즘에 관련된 이론을 습득하고, 문제해결 능력을 기르는 효과적인 교육방법 중 하나이다. 최근 교수-학습 과정에서 컴퓨터의 활용성을 고려할 때, 과제를 출제하고 평가, 피드백을 제공하는 과정이 자동화 된다면, 일련의 과정을 수행하는데 드는 시간과 비용, 그리고 노력의 절감이 가능할 것이다. 따라서 본 논문에서는 과제의 출제와 제출, 평가, 피드백을 제공할 수 있는 일련의 시스템을 설계하고 개발 하였다.

1. 서론

컴퓨터 프로그래밍 교육에 있어서 프로그래밍 과제는 자료구조 및 알고리즘에 관련된 이론을 습득하고, 문제해결 능력을 기르는데 가장 효과적인 교육방법 중 하나이다.

특히 최근 인터넷을 이용한 원격강의나 e-Learning이 보편화되는 추세를 감안할 때 수강생들이 제출한 과제물을 정확하고 공정하게 평가하는 것은 어떤 교육 내용보다 중요하다[1]. 하지만 많은 학생들이 제출하는 과제를 평가하고 피드백을 제공하는 것은 쉽지 않은 일이다. 평가자가 프로그램을 직접 실행시켜 볼 때 해당과목의 수강생이 증가할수록 많은 시간과 비용, 노력이 필요하기 때문이다.

이러한 시간과 비용을 줄이고 효율적으로 프로그래밍 과제를 평가하기 위한 시스템으로는 Online-Judge[2]와 ESPA[3]이 있다. 또한 과제가 아닌 프로그램의 평가만을 중점으로 하는 시스템도 존재한다. ACM-ICPC를 준비하는 학생들에게 2000 여개의 문제셋을 제공하는 Online-Judge Problem Set[4]이 스페인의 바야돌리드 대학에서 개발되어 운영중이고, 소프트웨어 개발 업체인 TopCoder[5]에서도 기업에서 요구하는 프로그래머의 능력 평가를 위해 온라인 프로그램 경진대회용 시스템을 운용하고 있다.

본 논문에서는 프로그래밍 과제에 대한 평가를 용이하게 하기 위하여 문제를 출제하고 입력에 대한 출력 결과를 확인하여 채점, 평가하여 학생 및 교수자에게 피드백을 제공하는 프로그래밍 과제 관리 및 평가 시스템을 설계하고 개발한다. 본 시스템은 학생이 과제를 제출하고 실시간으로 평가받을 수 있다. 또한 채점되는 과정을 확인할 수 있으며, 관리자는 과제 뿐만 아니라 과제가 출제되는 교과목을 관리할 수 있는 장점을 가지고 있다.

2. 시스템 설계

과제를 출제하고 채점하는 것은 일관성 있는 기준이 있어야 한다. 특별히 자료구조 및 알고리즘 과목의 프로그래밍 과제는 입력값에 해당하는 출력값이 정해져 있다. 따라서 학생들이 제출한 소스를 자동으로 컴파일하고 테스트 케이스를 입력하여 그 결과값을 비교하는 것만으로 정확한 평가가 가능하다.

출제자가 문제를 출제하고 테스트케이스를 등록하면 학생들은 문제를 해결하는 프로그램을 작성하여 이를 출제자에게 제출한다. 출제자는 학생들이 작성한 프로그램 소스 및 실행결과를 분석하여 평가하고, 적절한 피드백을 학생에게 제공하는 것으로 과제의 라이프 사이클은 끝난다.

즉, 출제 - 제출 - 평가 - 피드백으로 과제의 라이프 사이클을 나눌 수 있다. 따라서 설계의 중점은 이런 과제의 라이프 사이클을 자동화하고 전체 과제를 관리하는데 두었다.

과제를 등록하고 관리하는 사용자의 사용사례는 그림 1 과 같다.

2.1. 학생

학생에게 제공하는 시스템의 기능은 아래와 같다.

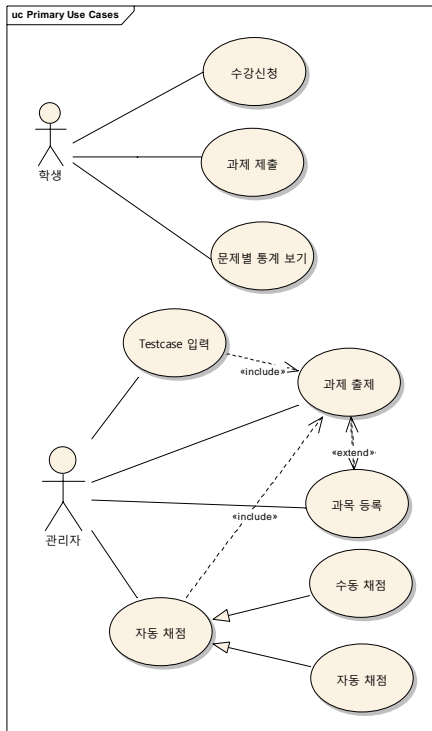
- 시스템에 로그인
- 프로그래밍 과제가 출제되는 과목에 수강신청
- 출제된 과제 확인
- 과제 풀이 및 제출
- 정답 및 오답 여부 확인 및 점수확인
- 전체 과제에 대한 자신의 현재 상태 확인

2.2. 관리자

관리자는 문제를 출제하는 출제자와 학생들의 점

수를 평가하는 평가자를 포함한다.
출제자를 위한 시스템의 기능은 아래와 같다.

- 시스템에 로그인
- 프로그래밍 과제를 출제하는 과목 개설
- 과제 출제
- 과제 제목, 과제 내용, 제출 마감일, 배점 및 감점요소, 실행 제한 시간, 첨부파일 및 테스트 케이스 등록
- 과제 제출현황 및 학생 점수 확인
- 과제 평가를 위한 점수별 정렬 기능
- 필요한 경우 평가 점수 수정



(그림 1) 시스템 사용 사례

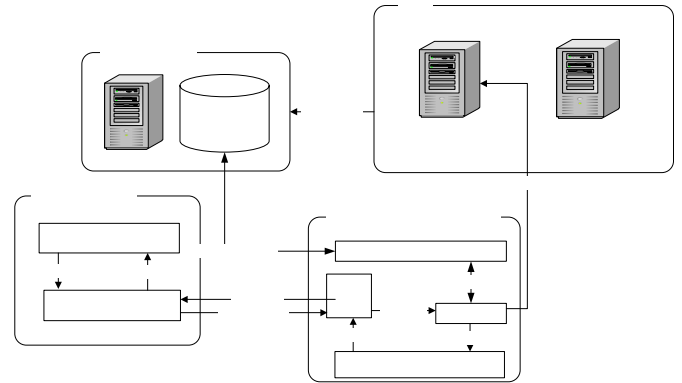
위와 같은 사용사례를 토대로 하여 구현한 시스템은 학생들이 제출하는 프로그래밍 과제에 대해서 자동으로 프로그램 소스를 컴파일하고 실행하여 실행 시간, 테스트 케이스 별 정답여부, 제출 기한에 따른 감점과 같은 객관적인 기준으로 점수를 평가한다.

또한, 문제 풀이 결과에 대한 다양한 통계를 제공함으로써 출제자 및 평가자는 피드백을 위한 기초 자료로 활용할 수 있고, 학생은 자신의 성취도를 판단할 수 있는 자료로 활용하여 학습능력을 향상시킬 수 있을 것이다.

3. 시스템 구현

전체적인 시스템은 리눅스(Ubuntu Linux Edge Eft)와 Apache2, Ruby On Rails 를 기반으로 작성되었고, 채점을 위한 컴파일 서버는 Java 1.6.0 을 이용하여 개발하였다. 사용자를 위한 클라이언트 프로그램은 Internet Explorer 나 Firefox 와 같은 웹 브라우저를 사용한다.

시스템에서 자료의 흐름은 아래 그림 2 와 같다.



(그림 2) 데이터 흐름도

3.1. 웹서버

웹 서버는 리눅스에서 운영되는 Apache2 를 기본으로 사용하였다. 이때 사용자의 입력값(파라미터)이 없는 정적인 페이지는 Apache2 가 단독으로 처리한다.

만약 사용자의 아이디와 비밀번호, 혹은 과제의 고유번호와 같은 파라미터가 전달된다면 그 페이지는 Ruby On Rails 에서 동적으로 웹페이지를 생성하여 Apache2 를 거쳐 사용자 클라이언트로 보내진다.

사용자의 파라미터를 처리해주는 Ruby On Rails[6]는 스크립트 언어인 Ruby 를 이용하여 개발된 웹 어플리케이션 개발용 프레임워크의 일종이다. Object-Relation Mapping 을 사용하여 데이터베이스의 테이블을 클래스로 매핑하고 클래스들 간의 관계를 설정하고, 클래스의 멤버 속성에 접근하는 것만으로 테이블에 자료를 쉽게 추가할 수 있는 장점이 있다. 또한, MVC(Model-View-Controller)패턴을 활용하여 데이터베이스 처리 부분을 담당하는 모델(Model), 데이터를 사용자에게 표시하는 일을 담당하는 뷰(View), 그리고 모델과 뷰를 제어하는 컨트롤러(Controller)를 통해 보다 빠르게 웹 어플리케이션을 개발 할 수 있다.

본 논문에서는 이러한 프레임워크를 이용해 시스템을 구현하였다. 예를 들어 과제 출제와 관련된 구현내용은 다음과 같다.

출제자가 과제를 출제할 때 클라이언트인 웹 브라우저에는 과제 이름과 과제의 내용, 제출 기한과 점수와 같은 데이터를 입력 받는 페이지가 표시된다. 출제자가 데이터를 모두 입력하고 확인 버튼을 누르면 사용자의 파라미터들은 해당 페이지와 연결된 Controller 로 전달된다. Controller 는 새로운 데이터를 데이터베이스에 입력하기 위해서 과제에 해당하는 테이블과 매핑된 클래스를 생성하고 각각의 멤버 속성에 사용자의 파라미터들을 대입한다. 이렇게 입력된 정보들은 Model 에서 검증과정을 거쳐 데이터베이스에 저장된다.

지금까지 출제된 과제의 목록을 보는 것도 마찬가지로 Controller 가 요청을 받으면 과제목록을 저장할 컨테이너를 만들고, Model 과 연동하여 데이터베이스에 있는 자료를 받아온다. 이렇게 저장된 자료들은 View 가 새로운 동적인 페이지를 만들 때 View 에게 전해져서 사용자가 과제 목록을 확인할 수 있게 한

다.

이런 일련의 과정들을 활용한다면 데이터를 받아 오기 위해서 복잡한 SQL 문장을 작성하지 않아도 되고, 모델에서 데이터에 대한 검증을 실시하기 때문에 잘못된 데이터가 들어갈 확률도 줄어들게 되어 결과적으로 개발에 필요한 시간을 단축 할 수 있다.

본 논문에서는 이와 같은 Ruby On Rails의 특징을 활용하여 보다 빠르게 웹 어플리케이션을 개발하였다.

3.2. 컴파일 서버

학생들이 제출한 소스를 컴파일 하고 채점하는 컴파일 서버는 Java 1.6.0 을 기반으로 하여 구현하였다.

학생이 과제를 제출할 때 제출한 소스 파일은 컴파일 서버의 특정 위치에 문제 번호 별로 폴더를 생성하여 학번.c 혹은 학번.cpp 형태로 저장된다.

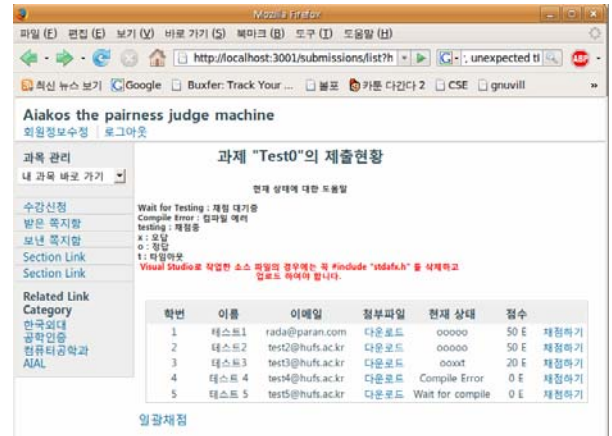
Ruby On Rails 에서 채점을 담당하는 Controller 는 평가자의 채점 요청이 있거나, 제출시 자동 채점 옵션에 활성화 되어 있는 경우 컴파일 서버에 TCP/IP 로 접속하여 [문제 번호][학번]이 포함된 패킷을 전달한다. 컴파일 서버가 패킷을 받으면 메시지 큐에 채점을 위해서 [문제 번호][학번]의 형태로 채점 요청을 저장하고 데이터베이스에 해당 과제의 채점 대기 중이라는 상태를 알려준다.

기존에 채점중이던 프로세스가 종료되면 소스 파일이 저장된 폴더에 접근하여 학생이 제출한 소스파일을 검색한다. 이때 확장자가 .c 이면 gcc 를, .cpp 이면 g++을 활용하여 소스파일을 컴파일한다. 컴파일 이 완료되면 채점을 위한 새로운 프로세스를 생성하고 프로세스의 표준 입출력을 파일로 redirection 한다. 프로그램이 실행되는 동안 데이터베이스에 저장되어 있는 테스트케이스를 입력하여 결과값을 얻어낸다. 이렇게 얻어낸 결과값을 데이터베이스에 저장된 결과값과 비교한 뒤 점수를 부여하는 것으로 채점은 완료된다.

만약에 실행시간 제한이 설정되어 있고, 컴파일 된 프로그램이 제한시간을 초과하여 실행되는 경우에는 채점을 위해 생성된 프로세스를 강제로 종료하고 데이터베이스에 제한시간초과로 예외를 처리할 수 있다.

표 1 과 그림 3 은 컴파일서버에서 각각의 배점이 10 점인 테스트케이스가 5 개 입력된 과제가 채점되는 과정을 보여준다. 표 1 에서 알 수 있듯이 채점이 진행되는 상황은 컴파일 서버에서 직접 데이터베이스에 실시간으로 입력되어 사용자가 과제의 채점 상황을 알 수 있게 한다.

컴파일 완료	컴파일 완료	0
채점 중	채점 중	0
제한시간초과	제한시간초과	0
런타임에러	런타임에러	0
부분정답	OXOOX	30
전체정답	OOOOO	50



(그림 3) 실행화면

3.3. 사용자 클라이언트

과제를 해결하는 프로그램을 작성한 뒤 소스 파일을 업로드 하는 것만으로 사용이 종료되는 학생들과 달리 평가자는 학생들의 점수를 한 화면에서 확인하고, 필요한 경우에는 학생들의 점수를 수정해야 하는 경우가 생긴다. 이때 학생들의 점수를 수정할 때 마다 저장 버튼을 눌러 페이지 전체를 다시 불러오는 것은 서버의 부하를 늘리고, 클라이언트의 편리한 사용을 방해한다. 따라서 클라이언트 프로그램은 수정된 부분만 서버에 전송하고 그 결과값을 얻어올 필요가 있다.

이런 기능은 AJAX[7]를 활용하여 구현하였다. AJAX(Asynchronous JavaScript And XML)는 SOAP, Javascript, XML, HTML, CSS 와 같은 기능을 사용하여 유연한 웹 클라이언트 개발에 자주 사용하고 있는 기술이다.

CSS 와 JavaScript 를 중점적으로 활용하여 학생들의 점수가 적혀있는 각각의 셀에 일련의 ID 를 부여한다, 각 셀의 수정 버튼이 클릭하면 CSS 와 JavaScript 를 활용하여 점수를 수정할 수 있는 입력창을 해당 셀에 표시하고, 점수를 수정하고 확인 버튼을 누르면 서버에 변경된 값을 전송하고 해당 부분을 업데이트 하는 방법으로 구현하였다.

4. 결론

매 학기마다 교과목에서 반복되는 과제 출제와 제출, 평가를 자동화하고, 과제를 관리 할 수 있는 시스템을 설계하고 구현하였다.

완성된 시스템은 웹을 이용하여 과제의 평가를 실시간으로 수행하여 학생들이 과제를 제출하는 즉시 자동으로 채점이 이루어지고 점수와 테스트케이스

<표 1> 컴파일 서버의 채점 상태

실행 결과	현재 상태	점수
제출 완료	제출 완료	0
컴파일 대기	컴파일 대기	0
컴파일 중	컴파일 중	0
컴파일 오류	컴파일 오류	0

통과여부를 알려준다. 한 화면에서 학생이 풀이해야 하는 다양한 과목의 과제를 표시하거나, 지금까지 제출한 과제의 점수를 확인 할 수 있고, 평가자에게는 수강생들의 점수를 관리할 수 있는 기능도 주어진다. 또한 중간 평가를 위해서 지금까지 제출된 과제들의 합산 점수를 정렬해서 보여주는 기능도 제공된다. 이런 기능들을 강의 과정에서 적절히 활용한다면 사람이 직접 채점하는 것 보다 효율적으로 과제를 채점하고 관리할 수 있을 것이다.

평가를 위해서 가장 먼저 해야 하는 일은 평가의 목적을 확인하는 것이다. 그리고 평가가 시행된 후 얻어진 결과는 그 특정 목적을 위하여 사용된다. 하지만 그 결과를 그대로 보고한다면 목적에 적합한 정보를 얻어내는 것이 어려운 경우가 생긴다. 따라서 상대 평가 혹은 절대 평가를 할 때 필요한 단순한 점수 외에도 과제의 난이도를 평가하는 등의 교수-학습 과정에서 필요한 다양한 통계자료를 제공할 필요가 있다. 또한 보다 공정한 평가를 위해서 프로그램 소스 코드의 표절 검출 기능이 필요할 것이다.

참고문헌

- [1] 지정훈, 우균, 조환규, “제한된 프로그램 소스 집합에서 표절 탐색을 위한 적응적 알고리즘,” 정보과학회논문지 : 소프트웨어 및 응용, 제 33 권 제 12 호, pp. 580-585, 2006.
- [2] Brenda Cheang , Andy Kurnia , Andrew Lim , Wee-Chong Oon, “On automated grading of programming assignments in an academic institution,” Computers & Education 41, pp. 121-131, 2003 .
- [3] <http://168.115.112.30:4608/math/espa/espa.html>
- [4] <http://online-judge.uva.es>
- [5] <http://topcoder.com>
- [6] Dave Thomas, and David Heinemeir Hansson, Agile Web Development with Rails 2/E, The Pragmatic Programmers, 2006.
- [7] 박형일 역, Foundations of Ajax, 사이텍미디어, 2006.
- [8] 장성순, 서선애, 이광근, “프로그램 유사성 검사기,” 2001 년 한국정보과학회 가을 학술발표논문집, Vol. 28, No.2 pp. 334-336, 2001.