

---

# CPUSim: CPU 스케줄링 알고리즘 교육을 지원하는 시뮬레이터

고정국\*

CPUSim: A Simulator supporting the education of CPU Scheduling Algorithms

Jeong-Gook Koh\*

## 요 약

운영체제는 추상적 개념과 기법들을 다루는 교과목이지만, 대부분 교재 위주의 이론 수업으로 진행되어 왔다. 이론 수업은 강의 내용에 대한 이해와 집중력 저하를 유발하므로 내용의 이해를 돕고 흥미를 유발하는 교육용 도구의 활용이 시도되었다. 본 논문에서는 CPU 스케줄링 알고리즘들의 동작을 시각적으로 보여주는 교육용 시뮬레이터를 설계하고 구현하였다. 구현된 시뮬레이터를 수업에 활용한 후 2010학년도와 2011학년도 수강생들의 학업 성취도를 측정하고 t-검증을 통해 두 집단의 CPU 스케줄링 알고리즘에 대한 시험문제 정답률 차이가 유의함을 밝혔다. 또한 시뮬레이터 활용에 대한 만족도와 관련분야 지식 습득에 대한 기여도 설문조사를 통해 시뮬레이터가 교과목에 대한 흥미를 유발하고 학습 내용에 대한 이해도를 증진시키는 교육용 도구로 유용하며, 이러한 수업 방식이 문제 해결능력 배양에 효과적임을 확인할 수 있었다.

## ABSTRACT

Operating Systems is a discipline which handles abstract concepts and techniques. However, most of OS courses have been textbook-oriented theoretical classes. Theoretical classes lead to the decline in the understanding of a lecture, and hurt their concentration. Many instructors have tried to make use of educational tools to help students understand lectures and arouse interests. This paper describes the design and implementation of a CPU scheduling simulator which shows the operation of process scheduling algorithms visually. The academic achievement evaluation for 2010's students and 2011's and t-test results show that the differences of the correct answer ratio for the exam about CPU scheduling algorithms are meaningful. The survey shows that the simulator is useful as an educational tool which causes the interests and enhances the understanding of a lecture, this teaching method is effective to develop problem solving skills.

## 키워드

운영체제, CPU 스케줄링 알고리즘, 교육용 도구, 시뮬레이터

## Key word

Operating system, CPU scheduling algorithm, educational tool, simulator

---

\* 정회원 : 동명대학교 컴퓨터공학과(교신저자, jgkoh@tu.ac.kr)

접수일자 : 2012. 02. 28

심사완료일자 : 2012. 03. 16

**Open Access** <http://dx.doi.org/10.6109/jkiice.2012.16.4.835>

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## I. 서 론

컴퓨터 분야의 교육 활동은 이론과 실습을 모두 포함하는데, 실습은 이론 수업에서 학습한 내용에 대한 이해를 돕고 활용능력 배양을 위해 필요하다. 수업 현장에서도 강의 시간에 이론적인 내용만 설명하는 것보다 관련된 실습이 추가되면 학생들의 이해도가 훨씬 높아진다[1]. 이론 수업은 시간이 경과하면서 집중도가 저하되고 지루함이 증대되지만, 실습을 병행하면 흥미를 잃지 않고 집중도를 유지할 수 있다[2]. IEEE Computer Society와 ACM의 합동 태스크 포스팀도 “컴퓨터 분야의 교육 과정은 이론과 실습을 병행하도록 편성해야 한다”라고 권고하고 있다[3].

한편 운영체제는 컴퓨터 시스템의 자원들을 효율적으로 관리하고 편리한 사용 환경을 제공하는 대표적인 시스템 소프트웨어이다[4,5]. 운영체제 강좌에서는 구성 요소의 동작 원리와 개념, 기법 등을 다루지만 운영체제의 동작을 직접 확인하기 어려워 대부분 이론 수업을 진행하고 있다. 새로운 개념이나 내용을 학습할 때 조작 가능한 도구의 활용이 바람직한데[6], 이론 강좌는 강의 내용에 대한 이해도와 집중력 저하를 초래하여 재미없는 과목으로 인식하는 경향이 나타난다. 따라서 멀티미디어 자료나 교육용 도구를 수업에 활용하여 흥미를 유발하고 강의 내용에 대한 이해도를 증진시키려는 시도가 다양하게 나타나고 있다[7].

본 논문에서는 학부 운영체제 강좌에서 다루는 CPU 스케줄링 알고리즘들에 대한 이해도 증진을 위해 프로세스 스케줄링 과정을 시각적으로 보여주는 시뮬레이터(CPUSim: CPU scheduling Simulator)를 설계하고 구현하였으며, 이를 수업에 활용하여 교육용 도구로 유용함을 확인하였다.

본 논문의 구조는 다음과 같다. 2장에서는 운영체제 교과목의 강의 개선 시도와 대표적인 CPU 스케줄링 알고리즘을 설명한다. 3장에서는 교육용 시뮬레이터의 필요성과 CPUSim의 기능 설계에 대해 기술한다. 4장에서는 CPUSim의 구현 내역과 기존 CPU 스케줄링 시뮬레이터들과 CPUSim의 특징을 비교한다. 5장에서는 수업에 CPUSim을 활용한 내용과 활용 효과를 기술한다. 마지막으로 6장에서는 본 논문의 결론 및 향후 연구방향을 제시한다.

## II. 본 론

## 2.1. 운영체제 교과목의 강의개선 시도

운영체제 강좌에서는 복잡하고 추상적인 개념과 처리 기법 등을 다양하게 다루지만 많은 강좌들이 이론 위주로 진행됨에 따라 교과목에 대한 관심과 흥미가 저하되는 현상이 나타난다. 이러한 문제점을 해결하기 위해 수업 시간에 학습 내용에 대한 흥미를 유발하고 이해도를 증대시키기 위한 다양한 방법들이 시도되어 왔다. 운영체제 분야의 강의 개선 방법들은 개인이나 팀 단위로 운영체제 기능 등을 구현하는 단기 프로젝트 수행[8-10], 교육용 운영체제나 실제 운영체제에 특정 기능을 구현하는 소스 코드 수정[11-13], 그리고 추상적인 개념의 이해를 돕는 교육용 시뮬레이터의 활용으로 구분할 수 있다[14].

한편, 교육용 시뮬레이터는 다양한 개념과 동작 방식을 체험할 수 있는 범용 시뮬레이터와 특정 기능만을 체험할 수 있는 단일 용도 시뮬레이터로 구분할 수 있다. 범용 시뮬레이터는 다양한 기능을 제공하므로 활용도는 높지만 시뮬레이터 개발 기간이 길고 개발 비용이 많이 소요되는 단점이 있다. 그러나 단일 용도 시뮬레이터는 특정 분야에 특화된 기능을 제공하므로 범용 시뮬레이터에 비해 활용 범위가 제한되지만, 개발 기간이 짧고 개발 비용도 적게 소요되는 장점이 있다. 대표적인 범용 시뮬레이터는 OSP[3], SOsim[14], RCOS.java[15] 등이며, 단일 용도 시뮬레이터는 BACI[16], MOSS[17], Alg\_OS[18], OSM[19], Robbins[20] 등이 있다.

## 2.2. CPU 스케줄링 알고리즘

컴퓨터 시스템의 모든 자원들은 사용 전에 스케줄링되어야 하는데, CPU도 시스템 자원의 일종이다. 프로세스가 작업을 수행하려면 CPU를 할당받아야 하는데, 어떤 프로세스에게 언제 CPU를 할당할 것인지 결정하는 작업이 CPU 스케줄링이다. CPU 스케줄링 알고리즘은 임의의 프로세스가 실행중인 프로세스를 중지시키고 CPU를 차지할 수 있는지 여부에 따라 선점과 비선점 기법으로 구분한다. 대표적인 CPU 스케줄링 알고리즘의 종류는 표 1과 같다[21].

표 1. CPU 스케줄링 알고리즘들의 종류  
Table. 1 The sorts of CPU scheduling algorithms

항목 종류	스케줄링 방법	특징	방식
FCFS	시스템에 들어온 순서대로 처리함	<ul style="list-style-type: none"> <li>간단하고 공평함</li> <li>처리시간 예측 가능</li> </ul>	비선점
SJF	실행시간이 가장 짧은 작업을 먼저 처리함	<ul style="list-style-type: none"> <li>짧은 작업에 유리함</li> <li>긴 작업은 평균 대기 시간이 길어짐</li> </ul>	비선점
SRT	실행 중에 더 짧은 작업이 존재하면 먼저 처리함	<ul style="list-style-type: none"> <li>실행시간의 지속적 추적 필요</li> <li>긴 작업은 SJF보다 대기시간이 길어짐</li> </ul>	선점
HRN	긴 작업과 짧은 작업간 지나친 불평등을 보완함	<ul style="list-style-type: none"> <li>응답률=(대기시간+서비스시간)/서비스시간</li> </ul>	비선점
라운드 로빈 (RR)	FIFO 방식의 변형, 일정한 CPU 시간을 할당함	<ul style="list-style-type: none"> <li>큰 할당시간: FCFS와 유사함</li> <li>작은 할당시간: 잦은 문맥교환 발생</li> </ul>	선점
우선 순위	우선순위가 높은 순서대로 처리함	<ul style="list-style-type: none"> <li>고정적 우선순위</li> <li>가변적 우선순위</li> </ul>	비선점

### III. CPUSim의 설계

#### 3.1. 교육용 시뮬레이터의 필요성

이론 수업에서는 교수자가 학생들에게 지식을 일방적으로 전달하는 수업 방식이 많이 사용되는데, 이 방식은 학생의 독립적인 사고와 상호작용을 저해하므로 교과목에 대한 흥미와 강의 내용에 대한 이해도를 저하시키는 문제가 있다[7]. 특히 운영체제처럼 추상적인 개념을 많이 다루는 교과목은 이러한 상황이 더욱 악화될 수 있다.

이러한 문제점을 해결하려면 수업의 중심을 교수자로부터 학생으로 옮겨야 한다[22]. 즉 교수자는 학생들에게 CPU 스케줄링 알고리즘들을 이론적으로 설명하고 교육용 시뮬레이터를 활용하여 처리 과정을 시연한다. 그리고 학생들은 시뮬레이터를 이용하여 직접 다양한 상황을 설정하고 알고리즘의 동작 과정과 실행 결과를 확인하며, 이들을 서로 비교해 봄으로써 CPU 스케줄링 알고리즘들에 대한 올바른 이해와 흥미를 유발하게

된다. 따라서 교육용 도구로 활용할 수 있는 CPU 스케줄링 시뮬레이터를 개발할 필요가 있다.

#### 3.2. CPUSim의 기능 설계

본 논문에서는 CPUSim의 형태를 단일 용도 시뮬레이터로 선정하고, 이를 활용하여 다양한 상황을 설정하고 알고리즘의 동작을 모의 실험하는 효과적인 교육 환경을 제공하고자 한다. CPUSim은 그림 1과 같이 환경 설정부, 알고리즘부, 결과 출력부의 세 부분으로 구성된다.

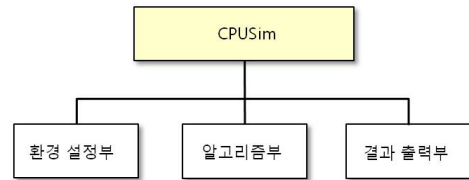


그림 1. CPUSim의 구성  
Fig. 1 The construction of CPUSim

##### 3.2.1. 환경 설정부

환경 설정부는 CPUSim의 구동에 필요한 시뮬레이션 매개변수 설정 기능을 제공한다. 설정가능한 매개변수들은 프로세스 생성 방법과 생성할 프로세스 수이다. ‘자동’을 선택한 후 프로세스 수를 입력하면 설정된 수 만큼 프로세스 관련 정보(프로세스명, 도착시간, 서비스 시간, 우선순위)들이 생성된다. ‘수동’을 선택하면 사용자가 해당 정보를 직접 입력한다. 그리고 RR 알고리즘을 선택할 때에는 프로세스에게 할당할 CPU 시간을 추가적으로 입력한다. 생성된 정보들은 프로세스명을 기준으로 정렬되어 CPUSim의 중앙 상단부에 표시된다. 그리고 매개변수의 값을 재설정할 수 있도록 ‘재설정’ 기능도 제공한다. 한편 프로그램을 처음 사용하면 대부분 어려움을 겪게 되므로 CPUSim에서는 도움말 기능을 제공한다.

##### 3.2.2. 알고리즘부

CPUSim은 운영체제 교재에 수록되어 있는 대표적인 6가지 알고리즘(FCFS, HRN, SJF, SRT, 우선순위, 라운드 로빈)을 지원한다. 본 논문에서는 우선순위 알고리즘을 새로운 프로세스가 준비 큐가 도착할 때 실행중인 프로세스와 우선순위를 비교하여 새로 도착한 프로세스

가 더 높으면 CPU를 선점시키는 방식으로 설계하였다. 사용자가 매개변수의 설정을 마친 후 알고리즘을 선택하고 ‘실행’ 버튼을 누르면 시뮬레이터가 구동된다.

### 3.2.3. 결과 출력부

결과 출력부는 설정된 프로세스 관련 정보에 근거하여 알고리즘의 처리 과정을 단계별로 보여준다. 즉, 프로세스들의 CPU 사용시간을 간트 차트로 출력한 후 해당 알고리즘의 성능 수치(평균 대기시간과 평균 반환시간)를 표시한다. 한편 동일한 상황에서 실행된 알고리즘들의 성능 수치가 누적되어 표시되므로 알고리즘들 성능 비교가 용이하다.

## IV. CPUSim의 구현

### 4.1. 구현 환경

본 논문에서는 CPUSim의 개발 언어로 Java를 선정하고, 개발 환경으로 Eclipse와 JDK 1.6.0\_23을 사용하였다. 그래픽 사용자 인터페이스의 구현을 위해 Swing을 사용하였고, 프로세스의 CPU 사용 시간은 JCommon과 JFreeChart 클래스 라이브러리를 활용하여 간트 차트로 나타내었다.

### 4.2. 구현 내역

#### 4.2.1. 사용자 인터페이스

CPUSim을 구동하면 그림 2의 사용자 인터페이스가 나타나며, 매개변수 설정과 결과 출력이 한 화면에서 이루어진다.



그림 2. CPUSim의 사용자 인터페이스  
Fig. 2 The user interface of CPUSim

사용자 인터페이스의 부분별 용도는 다음과 같다. 왼쪽 상단부는 매개변수를 설정하며, 중앙 상단부는 생성된 프로세스 정보를 보여준다. 왼쪽 하단부는 알고리즘을 선택하며, 우측 상단부는 프로세스 스케줄링 결과를 간트 차트로 표시한다. 우측 하단부는 실행된 알고리즘의 성능 수치를 표시한다. 중앙 하단부는 알고리즘의 성능비교 정보를 출력하며, CPUSim의 동작 제어 버튼이 배치된다. CPUSim의 동작 과정은 그림 3과 같다.

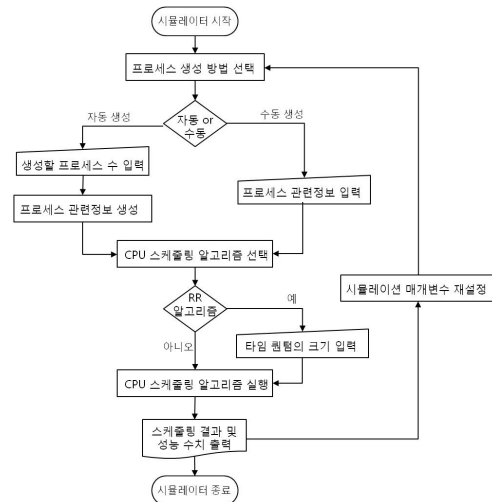


그림 3. CPUSim의 동작 과정  
Fig. 3 The operation process of CPUSim

#### 4.2.2. 환경 설정부

‘자동’ 생성을 선택하면 그림 4(a)와 같이 프로세스 수만 입력하고, ‘수동’ 생성을 선택하면 그림 4(b)와 같이 프로세스 관련 정보를 직접 입력한다.

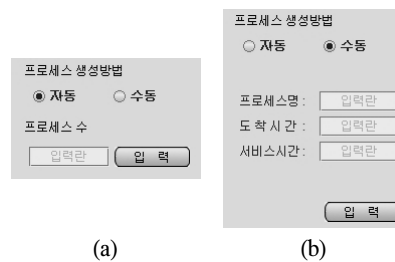


그림 4. 프로세스 관련 정보의 생성  
(a) 자동 생성 (b) 수동 생성

Fig. 4 The generation of process related informations  
(a) Auto generation (b) Manual generation

입력란이나 기능 버튼에 대한 설명이 필요할 때 마우스 포인터를 해당 부분에 위치시키면 그림 5와 같은 도움말이 나타나며, 마우스 포인터를 다른 곳으로 옮기면 도움말이 사라진다.



그림 5. 도움말 기능  
Fig. 5 The Help function

매개변수 설정이 완료되면 알고리즘 선택을 변경하더라도 기존 설정값에 근거하여 시뮬레이션을 진행하며, '재설정' 기능을 이용하면 설정값을 변경할 수 있다.

#### 4.2.3. 알고리즘부

알고리즘부는 알고리즘 목록과 선택 버튼을 그림 6과 같이 제공하며, 시인성 향상을 위해 선택된 알고리즘은 빨간색으로 표시된다. 다른 알고리즘을 선택하면 다시 검은색으로 변경된다.

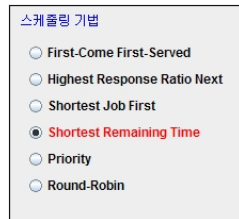


그림 6. 알고리즘 선택 기능  
Fig. 6 The Selection function for algorithms

#### 4.2.4. 결과 출력부

스케줄링 결과는 간트 차트로 표시되며, 알고리즘의 성능 수치가 출력된다. '자세히' 버튼은 성능 수치의 산출 내역을 보여준다. 그리고 그동안 실행된 알고리즘들의 성능 수치가 연속적으로 표시된다. 그림 7은 프로세스가 10개, 시간 할당량이 2일 때 라운드 로빈 알고리즘의 프로세스 스케줄링 결과와 2가지 성능 수치 정보를 보여준다.

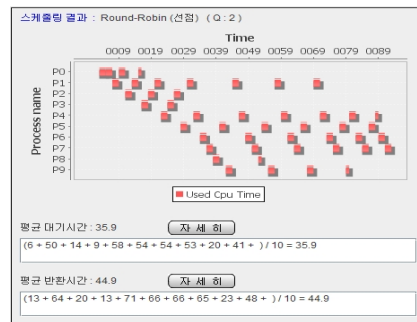


그림 7. 라운드 로빈 알고리즘의 스케줄링 결과  
Fig. 7 The scheduling result of RR algorithm

#### 4.3. 기존 연구사례와의 비교

기존의 CPU 스케줄링 시뮬레이터와 CPUSim의 특징을 비교한 결과는 표 2와 같다.

표 2. 기존 시뮬레이터들과 CPUSim의 비교  
Table. 2 The comparison of the existing simulators and CPUSim

종류	SOsim	Robbins	OSM	CPUSim
프로그램 실행 형태	독립 실행형	독립실행형+ 웹브라우저	DB 연동형	독립 실행형
활용 범위	범용	단일 용도	단일 용도	단일 용도
알고리즘수	2가지	6가지	4가지	6가지
상황 설정	수동	수동	수동	자동/수동
정보 제공	과정 위주	결과 위주	과정 위주	과정 위주
화면 구성	단일 화면	다중 화면	다중 화면	단일 화면
사용 난이도	낮음	높음	중간	낮음
도움말 제공	×	×	×	○
성능 비교	×	×	○	○
실행환경구축	쉬움	쉬움	어려움	쉬움
학습과정제공	×	×	○	×

SOsim[8]은 다중 프로그래밍, 프로세스 관리 및 스케줄링, 메모리 관리 기능을 제공하는 독립 실행형 범용 시뮬레이터이다. 2가지 유형의 알고리즘(비선점형, 선점형)을 지원하며, 로그 저장 기능을 제공한다. 그러나 프로세스 스케줄링에 한정하면 지원 알고리즘 수와 제공 기능이 미약하다.

Robbins[20]는 웹 브라우저와 독립 실행이 가능한 단일 용도 시뮬레이터로서 6가지 알고리즘(FCFS, SJF, 선

점형 SJF, SJFA, RR, 우선순위)을 지원한다. 메인 화면에서 메뉴를 선택하면 실행 결과가 별도 화면으로 제공된다. 매개변수 설정과 스케줄링 결과의 저장 기능을 제공하지만, 화면 구성이 복잡하고 사용이 어렵다. 스케줄링 정보는 결과 위주로 제공한다.

OSM[19]은 DB를 활용하는 단일 용도 시뮬레이터로서, 4가지 알고리즘(FCFS, SJF, 우선순위, RR)을 지원한다. CPU 스케줄러에 대한 이론적 설명을 포함하는 3단계 학습 과정을 제공하며, 알고리즘의 처리 과정을 시각적으로 보여준다. 처리 결과를 DB에 저장하며 결과값을 이용하여 알고리즘간 성능을 비교한다. DB를 활용하므로 실행환경 구축과 시뮬레이터의 독립적 실행이 어렵다.

CPUSim은 독립 실행형 단일 목적 시뮬레이터로서 6가지 알고리즘(FCFS, HRN, SJF, SRT, 우선순위, RR)을 지원한다. 매개변수의 자동/수동 설정이 가능하며 알고리즘의 처리 과정과 성능 수치 등을 하나의 화면에서 확인할 수 있다. 도움말 기능을 제공하며, 성능 정보가 누적되어 표시되므로 결과값의 저장없이 비교가 가능하다. 또한 DB를 사용하지 않기 때문에 시뮬레이터의 독립적 실행이 가능하지만, 이론적 지식 습득 후 CPUSim을 사용하므로 학습 과정은 제공하지 않는다.

## V. CPUSim의 활용 및 효과 분석

### 5.1. CPUSim의 활용

교수자는 CPU 스케줄링에 대한 올바른 이해와 문제 해결능력 배양을 위해 CPUSim을 활용하였다. 우선 CPU 스케줄링 알고리즘의 종류와 처리 방식 등을 설명하고 CPUSim을 활용하여 처리 과정을 시연하였다. 이론적 지식을 습득한 학생들은 실습실에서 CPUSim의 사용법을 익힌 후 이를 활용하여 알고리즘의 처리 과정과 결과를 확인하는 조별 활동을 수행하였다. 교수자는 조별 활동 후 학생들과 학습 내용에 대한 토론을 실시하고 퀴즈 시험을 통해 이해도를 점검하였다.

### 5.2. 활용 효과 분석

본 논문에서는 CPUSim의 활용 효과를 파악하기 위해 동일 교수자가 진행한 학부 운영체제 강좌를 대상으로 활용 이전과 이후의 수강생들의 학습능력 차이를 분

석하고 학업 성취도를 측정 후 t-검증을 하였다. t-검증은 두 모집단간 평균의 차이 유무를 판단하는 통계적 검증 방법이다[23].

우선 CPUSim 활용 이전인 2010학년도 수강생 33명은 비교 집단으로, CPUSim을 활용한 2011학년도 수강생 37명은 실험 집단으로 선정하였다. 두 집단의 학습능력에 대한 사전 동질성 검사를 위해 선수 교과목 이수정도, 학습한 프로그래밍 언어 수, 프로그래밍 능력을 분석 항목으로 선정하고 5점 척도로 측정하였다. 측정 결과에 대한 t-검증 결과는 표 3과 같이 학습 능력에 차이가 없는 것으로 나타났다.

표 3. 두 집단의 학습능력 비교

Table. 3 The comparison of the learning abilities

분석 항목	2010년		2011년		t값	p값
	평균	분산	평균	분산		
선수 교과목 이수정도	3.67	1.85	3.62	1.74	0.14	0.89
프로그래밍 언어 수	3.12	1.36	3.05	1.16	0.25	0.80
프로그래밍 능력	2.88	1.48	3.03	1.14	-0.54	0.59

\*p < 0.05

다음으로 사후 효과성 분석을 위해 시험문제 정답율을 평가 항목으로 선정하고, 중간고사에 CPU 스케줄링 알고리즘의 동작 과정과 성능 수치를 기술하는 문제를 3문항 출제하였다. 문제의 난이도 조절을 위해 알고리즘의 특성을 감안하여 종류는 다르게 선정하였다. 학업 성취도 분석 결과는 표 4와 같이 시험문제 정답율의 차이가 유의한 것으로 나타났다.

표 4. 두 집단의 정답률 비교

Table. 4 The comparison of the correct answer ratios

분석 항목	2010년		2011년		t값	p값
	평균	분산	평균	분산		
문제 정답율	0.692	393.50	0.848	153.51	-4.00	0.00**

\*\*p < 0.01

한편, 2011학년도 수강생을 대상으로 설문조사를 통해 시뮬레이터를 활용하는 수업 방식에 대한 만족도와

CPU 스케줄링 분야의 전공지식 습득에 대한 기여도를 파악하였다. 만족도 조사는 학기 말에 실시하였고, 조사 결과는 그림 8과 같이 수강생의 86.5%가 이러한 수업 방식에 만족하고 있음을 알 수 있다[24].

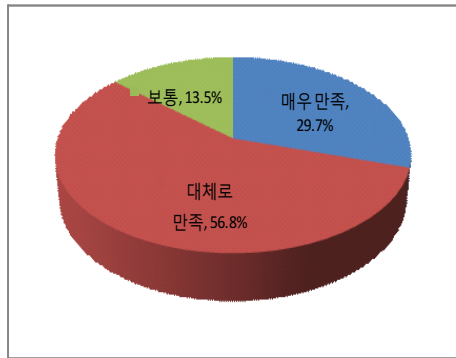


그림 8. 수업 방식에 대한 만족도  
Fig. 8 The satisfaction of the teaching method

지식 습득에 대한 기여도 조사는 해당 단원의 강의가 종료된 시점에 실시하였으며, 그림 9와 같이 수강생의 78.4%가 도움이 되었다고 응답하였다.

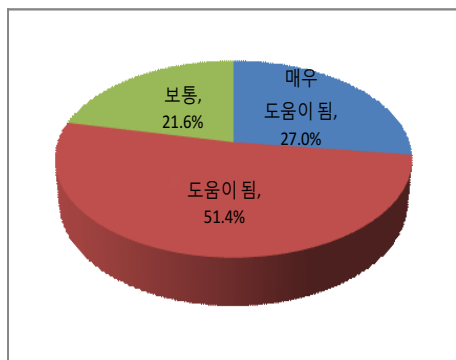


그림 9. 지식 습득에 대한 기여도  
Fig. 9 The contribution for knowledge acquisition

학업 성취도 측정과 설문조사를 통해 CPUSim이 수강생들에게 학습 내용을 올바르게 이해시키는 유용한 도구이며, 이러한 수업 방식이 문제 해결능력 배양에 효과적임을 확인할 수 있었다.

## VI. 결 론

본 논문에서는 CPU 스케줄링 알고리즘에 대한 수강생들의 이해도 증진을 위해 처리 과정을 시각적으로 보여주는 교육용 시뮬레이터인 CPUSim을 구현하였다. CPUSim을 수업에 적용한 후 시뮬레이터 활용 이전과 이후의 수강생을 대상으로 학업 성취도를 측정하고 t-검증을 실시하였다. 학업 성취도 분석 결과는 해당 항목의 차이가 유의한 것으로 나타났다. 또한, 설문조사를 통해 시뮬레이터를 활용하는 수업 방식에 대한 만족도와 지식 습득에 대한 기여도를 파악한 결과 수강생의 86.5%가 만족하고, 78.4%가 도움이 되었다고 응답함에 따라 이러한 수업 방식이 학습 내용에 대한 이해도 증진과 문제 해결능력 배양에 효과적임을 확인할 수 있었다.

향후 CPUSim에 웹 기반의 학습 환경을 제공하고, 알고리즘의 처리 과정과 결과를 사용자가 입력한 후 정오를 확인하는 평가 기능을 추가하는 등 지속적으로 기능을 개선하면 활용 효과가 극대화될 것이다.

## 참고문헌

- [1] 김재천, 부재율, 소경희, 채선희, 예비·현직 교사를 위한 교육과정과 교육평가, 3판, 서울: 교육과학사, 2005.
- [2] 임규혁, 임웅, 학교학습 효과를 위한 교육심리학, 2판, 서울: 도서출판 학지사, 2010.
- [3] M. Kifer and S. Smolka, "OSP: An Environment for Operating System Projects," *Operating Systems Review*, vol. 24, no. 4, pp. 98-100, 1992.
- [4] 오상엽, 최현섭, 운영체제, 고양: 이한출판사, 2005.
- [5] 구용완, 운영체제, 고양: 이한출판사, 2002.
- [6] 안정은, 김영봉, 구성주의 학습이론에 기초한 3차원 자전거 박물관 학습 코스웨어, 한국콘텐츠학회 춘계종합학술대회논문집, 제5권, 제1호, pp. 19-23, 2007.
- [7] 박성익, 임철일, 이재경, 최정임, 교육방법의 교육공학적 이해, 서울: 교육과학사, 2007.
- [8] A. Downey, "Teaching experimental design in an operating systems class," Proc. of the 30<sup>th</sup> SIGCSE technical symposium on Computer Science Education,

- vol. 31, no. 1, pp. 316-320, 1999.
- [9] T. Wagner and E. Ressler, "A practical approach to reinforcing concepts systems in introductory operating systems class," Proc. of the 28<sup>th</sup> SIGCSE technical symposium on Computer Science Education, vol. 29, no. 1, pp. 44-47, 1997.
- [10] S. Ramakrishnan and A. M.Lancaster, "Operating Systems Projects: linking theory, practice and use," Proc. of the 24<sup>th</sup> SIGCSE technical symposium on Computer Science Education, vol. 25, no. 1, pp. 256-260, 1993.
- [11] A. Tanenbaum and A. Woodhull, *Operating Systems: Design and Implementation*, 2nd Ed., NJ: Prentice-Hall, 1997.
- [12] D. Comer, *Operating system design: The XINU approach*, NJ: Prentice-Hall, 1984.
- [13] W.Christopher, S.Procter and T.Anderson, "The Nachos Instructional Operating System," Proc. of the Winter 1993 USENIX Conference, pp. 481-489, 1993.
- [14] L. Maia and A. Pacheco, "A simulator supporting lectures on operating systems," Proc. of 33<sup>rd</sup> Annual Frontiers in Education, vol. 2(F2C), pp. 13-17, 2003.
- [15] D. Jones and A. Newman, "RCOS.java: A simulated operating system with animations," Proc. of Computer-Based Learning in Science Conference, 2001.
- [16] B. Bynum and T.Camp, Alfonso: A Mutual Exclusion Toolkit, [http://inside.mines.edu/~tcamp/baci/baci\\_index.html](http://inside.mines.edu/~tcamp/baci/baci_index.html).
- [17] R. Ontko, MOSS Memory Management Simulator User Guide, [http://www.ontko.com/moss/memory/user\\_guide.html](http://www.ontko.com/moss/memory/user_guide.html).
- [18] A. Garpis, "Design and Development of a Web-based Interactive Software Tool for Teaching Operating Systems," *Journal of Information Technology Education*, vol. 10, pp. 1-17, 2011.
- [19] 정성균, 이상곤, "CPU 스케줄링을 학습하는 운영체제 시뮬레이션 프로그램의 설계 및 구현," 한국멀티미디어학회논문지, 제14권, 제6호, pp. 449-461, 2011.
- [20] S. Robbins and K.A. Robbins, "Empirical Exploration in Undergraduate operating systems," Proc. of the 30<sup>th</sup> SIGCSE technical symposium on Computer Science Education, vol. 30, no. 1, pp. 311-315, 1999.
- [21] 최현섭, 김성진, 이강성, 김성동, 운영체제론, 고양: 이한출판사, 1997.
- [22] M. Ben-Ari, "Constructivism in Computer Science Education," Proc. of the 29<sup>th</sup> SIGCSE technical symposium on Computer Science Education, vol. 29, no. 1, pp. 257- 261, 1998.
- [23] 독립표본 T-검정, [http://survey.kiet.re.kr/help/TreeHelp/analysis\\_2page-9.html](http://survey.kiet.re.kr/help/TreeHelp/analysis_2page-9.html).
- [24] 고정국, "디스크 스케줄링 알고리즘을 위한 교육용 시뮬레이터의 설계 및 구현," 한국컴퓨터정보학회 논문지, 제16권, 제12호, pp. 131- 137, 2011.

#### 저자소개



고정국(Jeong-Gook Koh)

1999년 부산대학교 컴퓨터공학과  
공학박사

1999년~현재 동명대학교  
컴퓨터공학과 부교수

※ 관심분야 : 운영체제, 임베디드 시스템, 컴퓨터활용  
교육