

## 디스크 스케줄링 알고리즘을 위한 교육용 시뮬레이터의 설계 및 구현

고 정 국\*

### Design and Implementation of an educational simulator for disk scheduling algorithms

Jeong-Gook Koh\*

#### 요 약

운영체제 교과목에서는 운영체제의 구성 요소와 동작 원리에 관련된 복잡하고 추상적인 개념들을 많이 다루고 있으나, 운영체제에 대한 교육은 주로 교재 위주의 이론 수업으로 진행되어 왔다. 다양한 형태의 매체에 익숙한 학생들에게 이론 위주의 수업은 강의 내용에 대한 이해도와 집중력 저하를 유발하므로 강의 내용에 대한 이해를 돕고 흥미를 유발하기 위한 교육용 도구의 활용을 지속적으로 시도하고 있다. 본 논문에서는 디스크 스케줄링 알고리즘들의 입출력 요청 처리 과정을 시각적으로 보여주는 교육용 도구인 디스크 스케줄링 시뮬레이터를 설계하고 구현하였다. 구현된 시뮬레이터를 운영체제 수업에 활용한 후 수강생들의 학업 성취도를 측정하고 설문조사를 통해 디스크 스케줄링 시뮬레이터가 운영체제 교과목에 대한 흥미를 유발하고 강의 내용에 대한 이해도를 향상시키는 교육용 도구로 유용함을 보였다.

▶ Keyword : 운영체제, 디스크 스케줄링 알고리즘, 시뮬레이터, 교육용 툴

#### Abstract

Operating Systems is a discipline which handles complex and abstract concepts related to its components and the principles of how an operating system works. However, most of the OS courses have been textbook-oriented theoretical classes. For students who are familiar with various kinds of media, theoretical classes lead to a decline in the understanding of a lecture and difficulty concentrating. They have tried to make use of educational tools to help students understand a lecture and arouse their interests consistently. This paper describes the design and implementation of a disk scheduling simulator which shows the service processes of disk I/O requests visually. The disk scheduling simulator can be used for demonstrations of the disk scheduling algorithms. The results of the academic achievement evaluation and survey showed that a disk scheduling

---

• 제1저자 : 고정국 • 교신저자 : 고정국

• 투고일 : 2011. 08. 18, 심사일 : 2011. 09. 30, 게재확정일 : 2011. 10. 18.

\* 동명대학교 컴퓨터공학과(Dept. of Computer Engineering, Tongmyong University)

simulator is useful as an educational tool which causes the interests about operating systems and enhances the understanding of a lecture.

▶ Keyword : Operating systems, Disk scheduling algorithm, Simulator, Educational tool

## I. 서 론

컴퓨터 관련분야의 교육 활동은 이론과 실습을 모두 포함하는데, 이론 수업에서 학습한 개념에 대한 이해를 돕고 전문적인 기술 습득을 위해서는 실습이 필요하다. 수업 현장에서도 강의 시간에 이론적인 내용만 설명하는 것보다 이론에 관련된 실습이 추가될 경우 학생들의 이해도가 훨씬 높아진다 [1]. 특히 강의 집중도 측면에서 이론 수업은 시간이 경과되면 학생들의 집중도가 저하되고 지루함을 느끼지만, 이론과 실습이 병행되면 시간이 경과해도 지속적으로 흥미를 느끼고 집중도가 유지되는 경향이 있다[2]. 이런 측면을 감안하여 IEEE Computer Society와 ACM(Association for Computing Machinery)의 합동 태스크 포스트는 “컴퓨터 분야의 교육과정은 학생들이 이론과 실습을 병행하도록 편성해야 한다”고 권고하고 있다[3].

한편 운영체제는 컴퓨터 시스템의 자원들을 효율적으로 관리하고 사용자에게 편의를 제공하여 하드웨어와 사용자간의 인터페이스를 담당하는 대표적인 시스템 프로그램이다[4]. 운영체제 교과목에서는 운영체제의 개념과 구성 요소, 동작 원리 및 제공 기능 등을 학습하지만 운영체제의 동작을 직접 눈으로 확인하기 어렵기 때문에 운영체제에 대한 교육은 주로 교재 위주의 이론 수업으로 진행되고 있다. 그러나 다양한 형태의 매체에 익숙한 학생들에게 이론 위주의 수업은 강의 내용에 대한 이해도와 집중력 저하를 초래하므로 운영체제 교과목이 지루하고 재미없는 과목으로 전락하는 경향이 나타나고 있다. 따라서 이러한 경향을 해소하기 위해 강의 시간에 멀티미디어 자료나 소프트웨어 등을 활용하여 흥미를 유발하고 강의 내용에 대한 이해도를 증진시키려는 연구가 다양하게 시도되고 있다[5].

본 논문에서는 학부 과정의 운영체제 교과목에서 디스크 스케줄링 알고리즘들의 입출력 요청 처리 과정을 시각적으로 보여주는 교육용 도구로 활용하기 위해 디스크 스케줄링 시뮬레이터를 설계하고 구현하였다.

본 논문의 구조는 다음과 같다. 2장에서는 운영체제 교과목의 강의개선 방법에 관한 연구와 대표적인 디스크 스케줄링 기법들을 설명한다. 3장에서는 디스크 스케줄링 시뮬레이터의 필요성과 기능 설계를 기술한다. 4장에서는 디스크 스케줄

링 시뮬레이터의 세부 구현 내역과 기존 디스크 스케줄링 시뮬레이터들과의 비교, 활용 효과에 대해 기술한다. 마지막으로 5장에서는 본 논문의 결론 및 향후 연구방향을 제시한다.

## II. 관련 연구

### 1. 강의개선 방법에 관한 연구

복잡하고 추상적인 개념들을 많이 다루는 운영체제 수업에서 학습 내용에 대한 흥미를 유발하고 이해를 돕기 위해 다양한 방법들이 많이 시도되어 왔다. 강의개선 방법들은 단기 프로젝트 수행, 소스 코드 수정, 그리고 시뮬레이터 활용의 세 가지로 분류할 수 있다.

첫 번째 유형은 프로세스간 통신, 동기화 프로그램, 시스템 유틸리티나 고전적인 운영체제 알고리즘 등을 구현하는 단기 프로젝트를 수행하는 것이다[6,7,8]. 단기 프로젝트는 구현이 용이하고 개발 시간도 짧지만 제공 기능이 제한적이므로 다양한 상황에서 활용하기 어렵다는 단점이 있다.

두 번째 유형은 Minix[9], Xinu[10] 등의 교육용 운영체제나 FreeBSD, Linux와 같은 운영체제에서 특정 기능을 구현하기 위해 소스 코드를 수정하는 것이다. 이 방법은 실제 운영체제에서 다양한 기능을 구현해 볼 수 있지만 컴퓨터 구조와 프로그래밍 언어, 운영체제에 대한 지식이 필요하므로 구현이 어렵고 개발 기간도 길다는 단점이 있다.

세 번째 유형은 병행성, 메모리 관리, CPU 스케줄링과 디스크 스케줄링 등의 운영체제 개념들에 대한 이해를 돕기 위해 시뮬레이터를 활용하는 것이다. 이 방법은 다양한 상황을 연출할 수 있기 때문에 단기 프로젝트에 비해 활용 범위가 넓고 소스 코드 수정에 비해 구현이 용이하고 개발 기간도 단축되는 장점이 있다.

한편, 교육용 시뮬레이터는 운영체제의 다양한 개념과 메커니즘을 체험해 볼 수 있는 범용 시뮬레이터와 메모리 관리나 CPU 스케줄링과 같은 특정 기능만을 체험할 수 있는 단일 목적 시뮬레이터로 구분할 수 있다. 대표적인 범용 시뮬레이터는 OSP[3], RCOS.java[11], SOsim[12] 등이며, 단일 목적 시뮬레이터에는 BACI[13], MOSS[14], Alg\_OS[15], OSM[16], Robbins[17], DiskSims[18] 등이 있다.

범용 시뮬레이터는 다양한 기능을 제공하므로 활용도는 높

지만 시뮬레이터 개발에 많은 시간과 비용이 소요되는 단점이 있다. 그러나 단일 목적 시뮬레이터는 특정 기능만 지원하므로 범용 시뮬레이터에 비해 활용 범위가 제한적이라는 단점이 있지만, 개발 시간과 비용이 절감되고 해당 분야에 특화된 기능을 제공할 수 있다는 장점이 있다.

## 2. 디스크 스케줄링 기법

디스크 스케줄링은 입출력 요청 큐에서 대기 중인 입출력 요청들의 디스크 상 위치를 고려하여 처리 순서를 결정한다. 디스크 스케줄링 기법들은 특성에 따라 탐색 시간 최적화 방법과 회전 지연시간 최적화 방법으로 구분할 수 있는데, 탐색 시간이 회전 지연시간에 비해 많은 시간을 소모하므로 디스크 스케줄링 기법들은 대부분 탐색 시간 최소화 목적을 두고 있다. 탐색 시간을 최소화하는 대표적인 디스크 스케줄링 기법들은 다음과 같다[4].

### 2.1 FCFS(First come first served)

먼저 도착한 요청이 우선적으로 서비스를 받기 때문에 공평성이 보장되지만, 탐색 패턴을 최적화하려는 시도가 없는 기법으로서 가장 간단한 형태이다.

### 2.2 SSTF(Shortest seek time first)

탐색 거리가 가장 짧은 입출력 요청을 먼저 처리하는 기법으로서, FCFS보다 처리량이 많고 평균 응답시간은 짧다. 그러나 특정 요청을 우선적으로 처리하기 때문에 응답시간의 편차가 큰 단점이 있다.

### 2.3 SCAN

SSTF와 유사하게 동작하지만 헤드 진행 방향으로 가장 가까운 거리에 있는 입출력 요청을 우선적으로 처리하며, 헤드가 마지막 트랙에 도달하면 진행 방향을 전환하기 때문에 엘리베이터 알고리즘이라고 부른다.

### 2.4 LOOK

SCAN 기법과 유사하게 동작하지만 헤드의 진행 방향으로 입출력 요청이 더 이상 없으면 진행 방향을 전환하여 반대 방향으로 이동하는 기법이다.

### 2.5 C-SCAN(Circular SCAN)

헤드를 바깥쪽 실린더에서 안쪽 실린더로 한쪽 방향으로만 이동하면서 입출력 요청을 처리하며, SCAN 기법의 불공평한 대기 시간을 개선한 기법이다.

### 2.6 C-LOOK(Circular LOOK)

C-SCAN 기법과 유사하지만 헤드 진행 방향으로 더 이상

입출력 요청이 없으면 헤드를 시작 위치로 이동한 후 다음 입출력 요청을 처리하는 기법이다.

## III. 디스크 스케줄링 시뮬레이터의 설계

본 절에서는 디스크 스케줄링 알고리즘의 입출력 요청 처리 과정을 시각적으로 보여주는 디스크 스케줄링 시뮬레이터의 필요성과 기능 설계에 대해 기술한다.

### 1. 디스크 스케줄링 시뮬레이터의 필요성

대부분의 이론 교과목들은 학생들이 교수자로부터 정보와 지식을 일방적으로 전달받는 수업 방식을 많이 사용하고 있다. 그러나 이러한 수업 방식은 학생의 독립적인 사고와 학생 간 상호작용을 저해하므로 교과목에 대한 흥미가 저하되고 강의 내용에 대한 이해도가 저하된다. 특히 복잡하고 추상적인 개념들을 많이 다루는 운영체제 교과목에서는 이러한 상황이 더욱 심화될 수 있다.

이러한 문제점을 해결하기 위해 수업의 중심을 교수자에서 학생으로 옮길 필요가 있다. 즉, 교수자는 학생들에게 디스크 스케줄링 알고리즘을 이론적으로 설명하고 시뮬레이터를 활용하여 처리 과정을 시연한다. 그리고 학생들이 시뮬레이터를 활용하여 직접 다양한 상황을 설정하고 알고리즘들의 처리 방식을 확인한다면 알고리즘에 대한 올바른 이해와 흥미 유발 효과도 거둘 수 있을 것이다. 따라서 운영체제 교과목의 교육용 도구로 활용할 수 있는 디스크 스케줄링 시뮬레이터를 개발할 필요가 있다.

### 2. 디스크 스케줄링 시뮬레이터의 기능 설계

본 논문에서는 학부 과정 운영체제 교과목의 디스크 스케줄링 알고리즘 교육용 도구의 형태를 범용 시뮬레이터에 비해 개발 시간이 짧고 해당 분야 학습에 필요한 기능만을 제공하는 단일 목적 시뮬레이터로 선정하였다. 디스크 스케줄링 시뮬레이터는 그림 1과 같이 환경 설정부, 스케줄링 알고리즘부, 결과 출력부의 세 부분으로 구성된다.

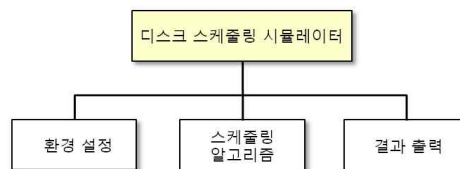


그림 1. 디스크 스케줄링 시뮬레이터의 구성  
Fig. 1. The organization of a disk scheduling simulator

### 2.1 환경 설정부

환경 설정부는 디스크 스케줄링 시뮬레이터의 구동에 필요한 시뮬레이션 매개변수들의 설정 기능을 제공한다. 설정가능한 시뮬레이션 매개변수는 디스크의 실린더 수, 디스크 헤드의 현재 위치, 디스크 헤드의 움직임 표시 속도, 디스크 입출력 요청 개수 및 생성 방법(자동 또는 수동), 디스크 헤드 이동 중 새로운 입출력 요청의 도착 여부, SCAN과 LOOK 기법 선택시 디스크 헤드의 이동 방향, 설정된 매개변수들의 유지 여부 등이다.

### 2.2 스케줄링 알고리즘부

디스크 스케줄링 시뮬레이터는 탐색 시간 최적화 방법에 속하는 대표적인 6가지 디스크 스케줄링 알고리즘(FCFS, SSTF, SCAN, LOOK, C-SCAN, C-LOOK)의 동작을 지원한다. 스케줄링 알고리즘부에서는 사용자가 6가지 알고리즘 중 하나를 선택한 후 시뮬레이터를 구동하는 기능을 제공한다.

### 2.3 결과 출력부

결과 출력부는 스케줄링 알고리즘의 입출력 요청 처리 과정과 결과를 시각화하여 보여준다. 그리고 환경 설정부의 '헤드 이동 속도' 조절 기능을 활용하면 차트 상에 표시되는 디스크 헤드의 궤적 표시 속도를 자유롭게 조절할 수 있다.



그림 2 디스크 스케줄링 시뮬레이터의 초기 화면  
Fig. 2. The main window of a disk scheduling simulator

일반적으로 프로그램을 처음 사용하는 경우 사용법에 익숙하지 못해 프로그램 사용에 어려움을 겪게 된다. 본 논문에서는 시뮬레이터를 쉽게 사용할 수 있도록 도움말 기능을 제공하고 있다. 사용자가 입력란에 마우스 포인터를 위치시키면 그림 3과 같은 도움말이 나타나며, 마우스 포인터를 다른 곳으로 옮기면 도움말이 사라진다.

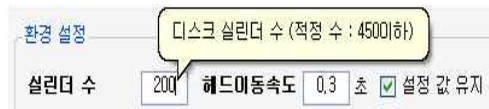


그림 3. 도움말 기능  
Fig. 3. The help function

## IV. 디스크 스케줄링 시뮬레이터의 구현

### 1. 구현 환경

본 논문에서는 닷넷 프레임워크 3.5와 C#.NET 언어를 사용하여 디스크 스케줄링 시뮬레이터를 구현하였으며, 디스크 헤드의 움직임은 마이크로소프트사의 닷넷 차트 컨트롤 [19]을 사용하여 표시하였다.

### 2. 구현 내역

#### 2.1 환경 설정부

디스크 스케줄링 시뮬레이터를 구동하면 그림 2와 같은 초기 화면이 나타나며, 사용자는 왼쪽 상단의 환경 설정부를 이용하여 시뮬레이션 매개변수들을 설정한 후 '확인' 버튼을 누른다. 그림 2는 디스크의 실린더 수가 200개, 디스크 헤드의 현재 위치는 53번 실린더, 헤드의 움직임 업데이트 주기는 0.3초, 디스크 입출력 요청은 자동으로 생성되며, 새로운 스케줄링 알고리즘을 선택할 때 이전에 설정된 매개변수 값들은 유지되도록 설정되어 있음을 보여준다.

#### 2.2 스케줄링 알고리즘부

사용자가 스케줄링 알고리즘을 선택한 후 '실행' 버튼을 누르면 선택된 알고리즘이 입출력 요청 큐에 대기하는 입출력 요청들을 처리한다. 구현된 시뮬레이터는 입출력 요청들에 대한 오프라인 스케줄링과 온라인 스케줄링 기능을 모두 지원하고 있다. 오프라인 스케줄링에서는 모든 입출력 요청들이 한꺼번에 도착하였다고 가정하고 입출력 요청들의 처리 순서를 결정한다. 온라인 스케줄링에서는 입출력 요청들을 처리하는 도중에 새로운 요청이 도착하더라도 기존 입출력 요청과 함께 이들의 처리 순서를 결정한다. 그림 4는 큐에 대기 중인 입출력 요청들을 처리하는 사이에 35번과 150번 실린더에 대한 입출력 요청들이 새로 추가되는 상황을 보여준다.

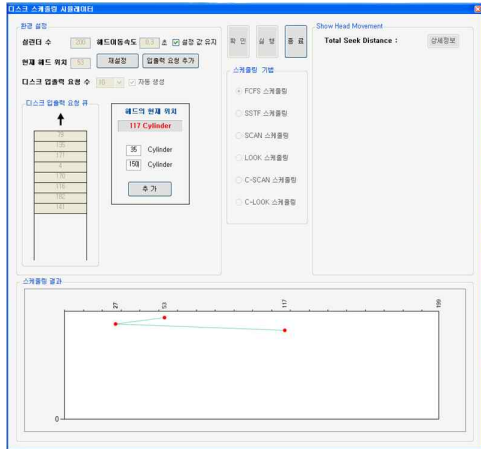


그림 4. 새로운 입출력 요청의 추가 기능  
Fig. 4. The function to add new I/O requests

2.3 결과 출력부

결과 출력부는 디스크 스케줄링 알고리즘의 입출력 요청 처리 과정을 직접 확인할 수 있도록 디스크 입출력 요청 큐의 변화, 디스크 헤드의 움직임, 그리고 디스크 헤드의 총 이동 거리를 보여준다. 한편 '상세 보기' 버튼을 누르면 입출력 요청별로 디스크 헤드의 이동 거리를 알 수 있다. 그림 5는 디스크 헤드가 안쪽으로 움직이고 있는 상황에서 SCAN 알고리즘이 10개의 디스크 입출력 요청을 처리하는 과정, 디스크 헤드의 총 이동거리와 입출력 요청별 헤드의 이동 거리를 보여준다.

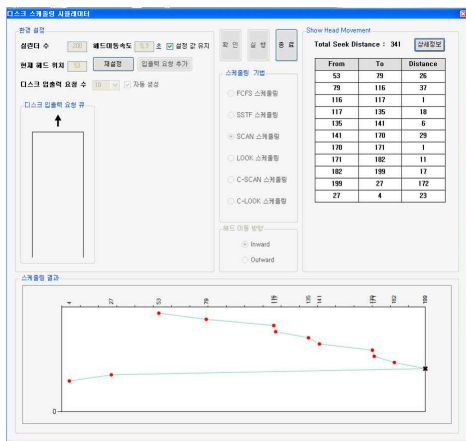


그림 5. 디스크 입출력 요청들의 시뮬레이션 결과  
Fig. 5. The simulation results of disk I/O requests

3. 비교 분석

본 절에서는 기존의 디스크 스케줄링 시뮬레이터[17,18]들과 본 논문에서 구현한 시뮬레이터를 비교한다.

Robbins[17]는 웹 브라우저 또는 독립 실행이 가능한 시뮬레이터로서 6가지 알고리즘(FCFS, SSTF, LOOK, C-LOOK, SCAN, C-SCAN)을 지원한다. 메인 화면의 메뉴를 선택할 때마다 결과 화면이 별도로 생성되며 다양한 시뮬레이션 매개변수 설정 기능과 처리 결과의 파일 저장 기능을 제공한다. 디스크 탐색 시간과 디스크 헤드의 움직임 등의 정보를 결과 위주로 제공하며 처리 과정은 보여주지 않는다.

DiskSims[18]는 GUI 인터페이스를 갖춘 독립 실행형 시뮬레이터로서 5가지 알고리즘(SSTF, LOOK, C-LOOK, SCAN, C-SCAN)을 지원한다. 화면은 다양한 시뮬레이션 매개변수 설정과 디스크 스케줄링 결과를 보여주는 2개의 탭으로 구성되어 있다. 시뮬레이션 관련 정보를 결과 위주로 보여주지만, 디스크 헤드의 움직임과 처리 과정은 나타내지 않는다.

기존의 시뮬레이터들이 시뮬레이션 결과에 중점을 둔 반면에 본 논문에서는 디스크 스케줄링 알고리즘의 동작을 확인할 수 있도록 처리 과정에 중점을 두고 있다. 구현된 시뮬레이터는 시뮬레이션 매개변수 설정과 처리 과정 확인이 용이하도록 GUI를 한 화면으로 구성하였고 독립 실행형이다. 6가지 알고리즘(FCFS, SSTF, LOOK, C-LOOK, SCAN, C-SCAN)을 지원하며, 알고리즘별로 디스크 입출력 요청 큐의 변화, 디스크 헤드의 움직임과 총 이동 거리, 입출력 요청별 헤드의 이동 거리 정보를 보여준다. 또한, 디스크 헤드의 이동 중 새로운 입출력 요청이 도착하더라도 기존의 입출력 요청들과 함께 처리하는 온라인 스케줄링 기능도 제공한다.

4. 디스크 스케줄링 시뮬레이터의 활용 효과

구현된 디스크 스케줄링 시뮬레이터는 운영체제 수업 시간에 교육용 도구로 활용하였으며, 수업 시간 외에 자가 학습을 통한 학습 효과 증진을 위해 수강생들에게 제공하였다. 그리고 시뮬레이터 활용 이전과 이후의 수강생들을 대상으로 1) 학업 성취도를 측정하고 2) 설문조사를 실시하여 활용 효과를 파악하였다.

학업 성취도 측정 대상은 시뮬레이터를 활용하지 않은 2010학년도 수강생 33명과 시뮬레이터를 활용한 2011학년도 수강생 37명이다. 평가 척도는 디스크 스케줄링 알고리즘에 관련된 시험 문제의 정답율과 시험 평균 성적이며, 측정 자료의 신뢰성과 타당성 확보를 위해 시험 문제 난이도를 유사하게 조정하였다. 표 1은 비교 결과로서 정답률은 23.6%, 평균 성적은 12.2% 상승하였다.

표 1. 시험 문제 정답률과 평균 성적  
Table 1. Correct answer ratio and average score

년도	시험 문제 정답률	평균 성적
2010	69.1%	74점
2011	85.4%	83점

설문 조사에서는 2011학년도 수강생을 대상으로 시뮬레이터를 활용하는 수업 방식에 대한 만족도와 해당 분야의 전공 지식 증대 기여도를 파악하였다. 수업 방식에 대한 만족도는 표 2와 같이 수강생의 86.5%가 만족하는 것으로 나타나 대체적으로 이러한 수업 방식에 만족하고 있음을 알 수 있었다.

표 2. 수업 방식에 대한 만족도 조사 결과  
Table 2. The research result of satisfaction for the teaching method

구분	빈도수	백분율
매우 만족한다	11	29.7%
대체로 만족한다	21	56.8%
보통이다	5	13.5%
불만족한다	0	0.0%
매우 불만족한다	0	0.0%
합 계	37	100%

표 3은 디스크 스케줄링 알고리즘 분야에 대한 지식 증대 기여도 조사 결과로서 전체 수강생의 73%가 도움이 되었다고 응답하였다.

표 3. 전공 지식 증대에 대한 기여도 조사 결과  
Table 3. The research result of the level of contribution for the knowledge growth

구분	빈도수	백분율
매우 도움이 되었다	10	27.0%
도움이 되었다	15	45.9%
보통이다	11	24.3%
별로 도움이 되지 않았다	1	2.7%
전혀 도움이 되지 않았다	0	0.0%
합 계	37	100%

학업 성취도 평가와 설문조사를 통해 수강생들이 이론적으로 학습한 내용을 시뮬레이터를 통해 직접 확인함으로써 운영 체제 교과목에 대한 흥미를 유발하고 학습 내용의 이해도 증진 효과도 거두고 있음을 확인할 수 있었다.

#### IV. 결 론

운영체제 교과목에서는 운영체제의 구성 요소와 동작 원리에 관련된 복잡하고 추상적인 개념들을 많이 소개하기 때문에

이들에 대한 이해를 돕고 흥미를 유발하기 위해 교육용 도구를 활용하려는 시도가 지속적으로 나타나고 있다.

본 논문에서는 디스크 스케줄링 알고리즘들의 입출력 요청 처리 과정을 시각적으로 보여주는 교육용 도구로 활용할 수 있는 디스크 스케줄링 시뮬레이터를 설계하고 구현하였다. 구현된 시뮬레이터를 운영체제 수업에 활용하고 활용 효과를 파악하기 위해 학업 성취도를 측정된 결과 시험 문제 정답률과 평균 성적이 각각 23.6%와 12.2% 만큼 상승하였다. 또한 설문조사 결과 수강생의 86.5%가 이러한 수업 방식에 만족하고 있으며, 73%가 디스크 스케줄링 알고리즘에 대한 이해도를 높이는데 도움이 되었다고 응답함에 따라 디스크 스케줄링 시뮬레이터가 교과목에 대한 흥미를 유발하고 학습 내용에 대한 이해도 증진 효과도 제공함을 확인할 수 있었다.

현재 구현된 디스크 스케줄링 시뮬레이터는 알고리즘별로 디스크 헤드의 움직임과 이동 거리만을 보여주고 있다. 앞으로 데이터 블록의 배치 설정 및 디스크 입출력 시간 산정 기능을 추가하는 등 시뮬레이터의 기능을 지속적으로 보완하고 웹과의 연동을 통해 언제 어디서든 편리하게 활용할 수 있도록 개선해 나갈 계획이다.

#### 참고문헌

- [1] Jae-chun Kim, Jae-yool Boo, Kyeong-hee Soh, Seon-hee Chae, "Curriculum and Educational Evaluation," 3rd Ed, Kyoyook Book, 2007.
- [2] Kyu-hyug Lim, Eung Lim, "Educational Psychology," 2nd Ed, Hakji publisher, 2008.
- [3] M. Kifer and S. Smolka, "OSP: An Environment for Operating System Projects," ACM SIGOPS Operating Systems Review, Vol. 24, No. 4, pp. 98-99, Oct. 1992.
- [4] Sang-yup Oh, Hyun-seop Choi, "Operating Systems," Ehan publishing com, 2005.
- [5] Seong-ik Park, Cheol-il Lim, Jae-kyung Lee, Jeong-im Choi, "Education technological Understanding of Instructional methods," Kyoyook Book, 2007.
- [6] A. Downey, "Teaching experimental design in an

operating systems class," Proc. of the 30th SIGCSE technical symposium on Computer science education, Vol. 31, No. 1, Mar. 1999.

[7] T. Wagner and E. Ressler, "A practical approach to reinforcing concepts in introductory operating systems," Proc. of the 28th SIGCSE technical symposium on Computer science education, Vol. 29, No. 1, Mar. 1997.

[8] S. Ramakrishnan and A. M. Lancaster, "Operating Systems Projects: linking theory, practice and use," Proc. of the 24th SIGCSE technical symposium on Computer science education, Vol. 25, No. 1, Mar. 1993.

[9] A. Tanenbaum and A. Woodhull, "Operating Systems: Design and Implementation," 2nd Ed, Prentice-Hall, 1997.

[10] D. Comer, "Operating system design : The XINU approach," Prentice-Hall, 1984.

[11] D. Jones and A. Newman, "RCOS.JAVA: A simulated operating system with animations," Proc. of CBLIS'2001, Brno, The Czech Republic, 2001.

[12] L. Maia and A. Pacheco, "A simulator supporting lectures on operating systems," Proc. of 33rd Annual Frontiers in Education, Vol. 3, Nov. 2003.

[13] B. Bynum and T. Camp, "After you, Alfonse: A Mutual Exclusion Toolkit," [http://inside.mines.edu/~tcamp/baci/baci\\_index.html](http://inside.mines.edu/~tcamp/baci/baci_index.html)

[14] 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)

[15] Aristogiannis Gampis, "Design and Development of a Web-based Interactive Software Tool for Teaching Operating Systems," Journal of Information Technology Education, Vol. 10, 2011.

[16] Seong-Kyun Jeong, Samuel Sangkon Lee, "Design and Implementation of Simulation Program for CPU Scheduling in Operating Systems," Journal of Korea Multimedia Society, Vol. 14, No. 3, pp. 449-461, Mar. 2011.

[17] Steven Robbins, "A Disk Head Scheduling Simulator," Proc. of the 35th SIGCSE technical symposium on Computer science education, Vol. 36,

No. 1, pp. 325-329, Mar. 2004.

[18] Ammar Muqaddas, "DiskSims: A Disk Scheduling Simulator tool with an intuitive GUI interface and command line functionality," <http://www.softpedia.com/get/Others/Miscellaneous/DiskSims.shtml>

[19] Microsoft Corp., "MS Chart controls for .NET Framework 3.5," <http://www.microsoft.com/downloads/ko-kr/details.aspx?displaylang=ko&FamilyID=130f7986-bf49-4fe5-9ca8-910ae6ea442c>

## 저 자 소개



### 고 정 국

1992 : 부산대학교 컴퓨터공학과  
공학사

1994 : 부산대학교 컴퓨터공학과  
공학석사

1999 : 부산대학교 컴퓨터공학과  
공학박사

1999~현재 : 동명대학교 컴퓨터공  
학과 부교수

관심분야 : 운영체제, 시스템 프로그  
래밍, 컴퓨터활용교육

Email : jgkoh@tu.ac.kr