

Development of Simulation App Tool for Understanding 8 Process Scheduling Policies

Kyong-ho Lee*

*Professor, Dept. of Information Communication and Software, Halla University, Kangwon, Korea

[Abstract]

In this study, an simulation app was developed as one of the methods to help learners better understand the eight process scheduling policies of multi-programming. In learning, an app in the form of a simulation should provide a realistic environment and allow learners to practice. To do this, the needs of the learners were investigated and analyzed, and the purpose was set, designed, and programmed based on the learners' understanding. And it was shown that the apps as a tool to simulate the created eight scheduling policies are performing well. In particular, it was shown that the problem of not having a step-by-step various diagram and explanation for step-by-step various inputs, which is a limitation of paper textbooks, can be solved using these tools.

▶ **Key words:** Simulation App, Process Scheduling, Educational App, App-Inventor, Operating System

[요 약]

본 연구에서는 학습자들이 다중 프로그래밍의 프로세스 스케줄링 정책 8가지를 잘 이해할 수 있도록 돕는 방법 중 하나로 시뮬레이션 앱을 개발하였다. 학습에서 시뮬레이션 형태의 앱은 실제와 유사한 환경을 제공해야 하며 학습자가 연습할 수 있도록 해야 한다. 이렇게 하기 위해 학습자들의 요구 사항을 조사 분석하고 학습자들의 이해를 중심으로 목적을 정하고, 설계하고 프로그래밍 하여, 8가지 스케줄링 정책을 시뮬레이션 할 수 있는 도구로서의 앱을 만들었고, 이것들이 잘 수행되고 있음을 보였다. 특히 지면 교재의 한계인 단계적 다양한 입력에 대한 단계적 다양한 도해와 그에 따른 설명을 갖지 못하는 문제를 이런 도구를 이용하여 해결할 수 있음을 보여주었다.

▶ **주제어:** 시뮬레이션 앱, 프로세스 스케줄링, 교육용 앱, 앱인벤터, 운영체제

I. Introduction

오늘날 컴퓨터들은 CPU 이용률을 극대화하기 위해 다중 프로그래밍 환경을 운영한다. 다중 프로그래밍 환경에서는 다수의 프로세스들이 제한된 자원들을 경쟁적으로 사용하기 때문에 경쟁을 조정하는 기능이 요구된다. 프로세스들의 경쟁을 조정하기 위한 다수의 알고리즘들이 있고, 운영체제 교재에서는 많은 프로세스 스케줄링 알고리즘들을 소개하고 있고, 알고리즘 마다 장단점이 있어 대부분이 사용되지만, 조금씩 다른 다양한 정책에 의해 수행되는 알고리즘이라 학습자들은 혼동을 하거나 이해에 어려움을 보이고 있다. 학습에서 교사는 학습자의 이해를 위한 지속적인 노력이 필요하며, 다양한 도구와 함께 학습의 매개체로서의 충분한 역할을 해야 한다[1]. 최근에 학습자들이 학습 내용을 잘 이해할 수 있도록 돕는 방법 중 하나로 수업 중에 앱을 많이 이용하고 있다. 학문과 정보통신소프트웨어 기술이 만나서 멋있는 미술 도구가 되기도 하며, 음악 도구가 되기도 하고, 과학 등의 수업에서 필요한 내용을 시각적으로 보여주기도 하고, 영화 제작도 하며, 보건과 예방 교육, 인문학, 공학, 농업 교육 등 유치원 교육에서 대학교육 또 다양한 사회에서의 교육까지 다양한 과목에서 다양한 도움을 주고 있다[2-5].

한국의 스마트폰 보급률은 2020년 현재 95%에 달한다[6]. 대학생들에게 보급률은 100%라고 말해도 문제가 없다. 또한 우리나라는 네트워크가 잘 발달되어 있어 학습 환경 어디서나 스마트폰을 사용하는데 문제가 없다. 젊은이들에게 스마트폰은 장난감이자, 타인과 연결하는 환경이고, 책도 되고, 할 일의 가이드도 되는 등 매우 친숙한 도구이다.

본 연구에서는 학습자들이 어려움을 겪는다고 판단되는 프로세스 스케줄링 알고리즘 8가지의 이해를 도울 수 있는 실습 도구로서의 앱을 구성하였다. 수업 시간에 이해를 했더라도 평가를 통해 확인해 보면 수업 시간에 이해한 내용들을 기억하지 못하는 상황을 볼 수 있다. 인간의 기억은 보고-듣기만 할 때 보다, 보고-듣고-실습할 때 더 많은 내용을 기억한다는 이론을 배경으로 하였다. 학습자들이 어려워하는 요구를 조사하고 여기에 교육자의 목적으로 학생들이 알아야 하는 바를 추가하여 수강자들이 지식을 습득하는데 도움이 될 8가지 프로세스 스케줄링을 실습할 수 있도록 하였다. 본 시뮬레이션 앱을 이용하여 8가지 프로세스 스케줄링 알고리즘의 이해를 원하는 학습자들은 따로 또는 같이 모여 논의하며, 의도를 반영하여 실습을 하며 8가지 프로세스 스케줄링의 지식을 습득할 수 있다.

II. Preliminaries

프로세스 스케줄링은 CPU 사용에 대한 프로세스들의 경쟁을 조정하는 기능이다. 컴퓨터들은 CPU 이용률을 극대화하기 위해 다중 프로그래밍 환경을 운영한다. 다중 프로그래밍 환경에서는 다수의 프로세스들이 제한된 자원들을 경쟁적으로 사용하기 때문에 경쟁을 조정하는 기능이 요구된다. 운영체제 안에는 프로세스들의 경쟁을 조정하기 위한 다양한 정책을 구성한다. 본 연구의 시작은 프로세스 스케줄링 수업을 하면서 정적 그림과 강의만으로 부족한 내용들을 바탕으로 학습자들에게 흥미와 이해 증진, 협력과 사고력 향상에 도움을 주기 위하여 교수자로서 인지한 필요성을 바탕으로 개발하고 이용하고자 한 것이다. 앱을 선택 이유는 모든 학습자가 소지하고 있는 스마트폰 환경을 생각한 것이다. 연구자가 2년간 정적 그림과 강의만으로 학습자들에게 전달한 효과를 평가를 이용하여 확인한 결과는 Fig. 1과 같다. y축은 빈도수이며, x축은 수강자의 답을 근거로 본 연구자가 분류한 것이다. 수업 시간에 이해를 했다고 질문을 하지 않지만 평가해 보면 이해도가 낮은 모습을 볼 수 있다.

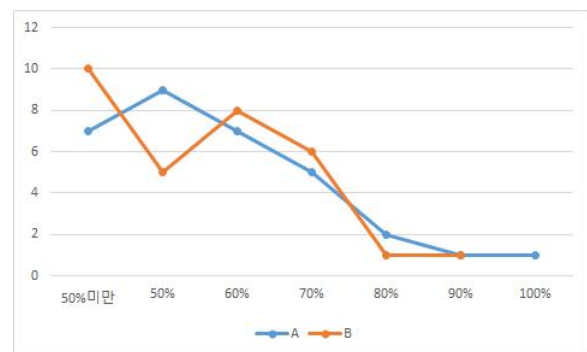


Fig. 1. Analysis table of effects delivered to learners only with static pictures and lectures

교육용 앱은 학습형태와 상호작용 방식에 따라 개인교수용, 반복학습형, 시뮬레이션형, 게임형, 자료제시형, 평가형, 도구형, 문제해결형으로 분류한다. 개인교수형은 새로운 개념이나 지식을 가르치기 위해 설명이나 안내를 하는 유형이며, 반복학습형은 일종의 보충이나 심화학습을 위해 반복적으로 지식을 확인하거나 기능을 연습하는 유형이며, 시뮬레이션은 실제와 유사한 환경을 제공하여 학습자가 연습할 수 있도록 하는 유형이며, 게임형은 목표, 규칙, 경쟁, 흥미, 도전, 호기심 등의 게임적인 요소를 첨가하여 학습을 하도록 설계된 유형이고, 자료제시형은 콘텐츠 안에 많은 분량의 자료를 저장하여 필요한 자료를 찾

아보며 학습하도록 한 유형이며, 평가형은 교수-학습 결과의 평가를 주된 내용으로 만든 유형이고, 도구형은 학습을 촉진시켜 주기위한 도구적 성격이며, 문제해결형은 학습자가 고차원적인 사고 기능을 활용하여 비구조화된 문제를 해결하도록 하는 유형이다[7-8]. 앱 스토어를 통해 교육용 앱들에 관하여 분석해 보면 대부분의 교육용 앱은 한자 학습이나 영어 단어 암기 등과 같은 반복 학습형과 선생님이 설명해주는 동영상이나 애니메이션 등과 같은 전통적인 튜토리얼 형태의 개인교수형이 많다. 조연정 등의 연구에서도 튜토리얼 형태와 개인 교수형 앱들이 교육용에서 대부분을 차지하고 있다[7]. 본 연구자가 확인한 바로도 대부분 교육용 앱은 보편적 활용을 위해 만든 것이며, 교수 현장에서 학습자의 필요성을 파악하고 반영한 것은 아니라고 판단되며, 특히 시뮬레이션형의 교육용 앱은 전체 관측용 앱 정도의 비교적 간단한 앱과 복잡한 임계 영역문제를 해결하기 위하여 만든 Test-and-Set 알고리즘 시뮬레이션 앱 정도를 볼 수 있었다[9].

본 연구에서 개발한 시뮬레이터 앱을 구성하기 위하여 사용한 환경은 앱 인벤터(App Inventor)이다. 앱 인벤터는 2010년 구글에서 만들어 공개되었고 현재는 MIT에서 관리되고 있는 웹 비주얼 프로그래밍 툴이다. 안드로이드 운영 체제용 앱을 쉽게 만들 수 있게 해준다. 시각 객체들을 드래그-앤-드롭하여 프로그램들을 만들 수 있다[10]. 그러나 간단한 프로그램들은 쉽게 만들 수 있지만 유저 인터페이스(user interface)가 시각화와 연결되며, 프로그램의 크기가 커질 경우 앱 인벤터 프로그램은 높은 난이도를 갖게 된다. 이 프로그래밍 툴은 프로그램 구성자가 해야 할 많은 시각적 일을 대신 구성해주는 구조로 이로 인하여 시각적 큰 부피를 갖고 있으며, 조금 큰 프로그램에서 코딩의 부피가 매우 커서 프로그램 구성자가 한 눈에 볼 수 있는 논리적 범위가 매우 좁아서 종종 문제가 발생되기도 한다. 또한 비주얼 툴의 특성으로 내부 코드는 보이지 않게 되어 있어, 규모가 조금 커지는 모듈에서 프로그래밍 언어 상 문제가 되지 않을 논리적 구조에서 실행 에러 문제를 일으키곤 하는데, 단지 로컬 변수 스코프(scope)를 변경하거나 논리적으로 모듈을 분할하면 해결되는 이해하기 어려운 실행 에러가 발생하기도 한다. 본 연구에서 협소한 코드 시야 문제는 코드 시야가 넓은 파이썬 언어로 같은 논리를 이상이 없도록 먼저 구성한 후 파이썬 코드를 가이드로 하여 앱 인벤터 프로그램을 구성하는 방법으로 해결하였고, 규모가 커지는 모듈에서 발생하는 이해하기 어려운 실행 에러 문제는 모듈을 분화하여 해결하였다.

III. Simulation App Design

인간의 기억은 보고-듣기만 할 때 보다, 보고-듣고-실습할 때 더 많은 내용을 기억한다[11]. 따라서 본 앱은 시각적 효과와 상호 작용적 효과를 얻기 위해 만들었다. 본 연구에서 만든 시뮬레이션 앱의 프로세스 스케줄링 수행 환경은 CPU 1개, 준비 상태 프로세스들의 대기 장소인 READY QUEUE는 스케줄링 정책에 따라 1개 또는 3개, 프로세스 상태는 준비 상태 / 실행 상태 / 대기 상태에서 스케줄링 정책에 맞추어 Fig. 2와 같이 변화한다. 여기에서 스케줄링 정책에 따라 프로세스에게 CPU를 할당 및 사용 중지하게 한다. 본 연구에서 구성한 스케줄링 정책은 선입선처리(First Come First Service) 스케줄링 / 최단 작업 선처리(Shortest Job First) 스케줄링 / 최소 잔여 시간 선처리(Shortest Remain Time First) 스케줄링 / 높은 응답 비율 다음(Highest Response Ratio Next) 스케줄링 / 우선순위(Priority) 스케줄링 / 라운드 로빈(Round Robin) 스케줄링 / 다단계 큐(Multi Level Queue) 스케줄링 / 다단계 되먹임 큐(Multi Level Feedback Queue) 스케줄링의 8가지를 구현하였다. 사용자는 시뮬레이션 정책에 따라 프로세스 생성 시간, CPU 실행 시간, 프로세스 우선 순위, CPU 할당 시간 등의 값을 입력해야 한다[12-13].

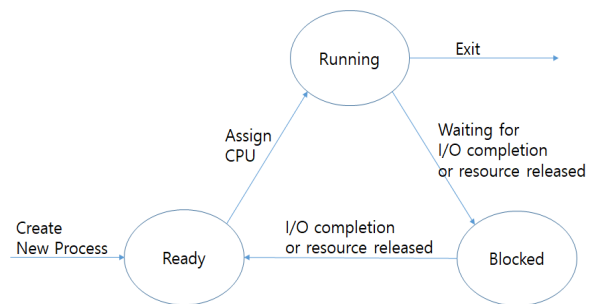


Fig. 2. Running Screen

교재에서 많이 사용하는 지식 전달 표현은 텍스트와 수식 그리고 도해이다. 그러나 텍스트와 수식은 정확성은 있으나 직관성이 부족하여 학습자는 자신 내부 지식을 이용해 해석을 하는 과정이 필요하여, 쉬운 이해에는 직관성이 다분한 도해가 흥미와 이해 증진 역할을 한다. 또한 기존 교재에는 도해가 학습자의 요구에 맞게 충분히 없을 수 있으며, 있더라도 저자의 관점에서 전달하고자 하는 모든 특징을 하나의 도해에 표현함으로써 도해가 복잡성을 내재하고 있기도 하여, 학습자의 단계적 요구를 충족주지 못하는 실정이다. 본 연구를 위해 학습자들과 파악한 시뮬레이터

의 요구사항은 '다양한 입력에 대한 실행 예가 시간의 변화에 따른 그림으로 볼 수 있으면 좋겠다. 쉬운 입력부터 여러 개의 실행 예를 볼 수 있으면 좋겠다. 이론을 이해한 후에 다양한 입력에 의한 CPU 이용률, 프로세스 처리율, 평균 반환 시간, 평균 대기 시간, 평균 응답 시간 및 프로세스별 반환 시간의 결과가 학습자가 계산한 것이 맞는지 확인할 수 있으면 좋겠다. 또 그렇게 결과가 나온 이유를 알 수 있으면 좋겠다. 스케줄링 별 입력의 종류가 다른 이유가 헛갈린다. 중도탈퇴 정책이 추가되면 상황 판단이 복잡해지는 것 같다. 대기 큐가 여러 개 있으면 혼란해 진다. 여러 개의 프로세스가 삽입되면 각 프로세스의 종료 시간이 혼동된다.' 등이다.

1. Design Direction

본 연구에서 학습자들에게 흥미와 이해의 증진, 사고력의 향상에 도움을 주기 위하여 갖추어야 할 조건으로 판단한 시뮬레이션 환경은 다음과 같은 점들이다. 위 학습자의 요구 사항에 부합되는 유저인터페이스(user interface) 환경을 구성한다(fig. 3이 결과물임). 이해를 위해 학습자가 입력을 해야 한다고 판단한 내용을 제외하면, 자동으로 구성되도록 하여 학습자가 다른 것에 신경 쓰다 분심이 생기지 않게 단순화 환경을 구성한다. 그래서 프로세스 ID는 프로세스의 구별을 위해 필요한 것이나 학습자가 넣는 것이 아닌 자동부여 되도록 하였고, 의미 이해의 과정에 중요한 행위로 프로세스 도착시간, CPU 사용 시간 등은 학습자가 넣도록 하되 스케줄링 정책 별로 합당한 범위만 입력하도록 입력 범위가 제한되게 구성해 주었다. 학습자의 현 이해 단계에서부터 수행해 보고, 또 요구되는 수준까지 다양한 입력을 할 수 있는 환경을 구성한다. 그래서 프로세스 1개만 넣고 수행해보고 그 상황에서 프로세스를 몇 개 더 추가하여 단계별로 수행해 볼 수 있다. 시뮬레이션 중간에 모두 취소하고 처음부터 시뮬레이션을 할 수도 있다. 키보드 없는 스마트폰 환경을 고려하여 단순한 선택과 같이 편리한 입력 환경을 구성한다. 그래서 스피너(Spinner) 기능을 이용하여 편리한 입력을 수행하도록 하였다. 입력 값은 그 상황에 맞는 범위에서 선택되도록 내부에서 판단하여 구성하고 구성된 내용으로 선택하도록 한다. 이를 위해 프로그램 내부에서는 매 단계마다 스피너 리스트들을 재구성해 준다. 이렇게 하여 학습자들이 프로세스들을 동일 시간에 2개 이상 발생될 수 없도록 하였고, 프로세스 간 사이는 연속적 또는 간격이 생길 수 있도록 하였다. 수업을 통해 각 스케줄링들을 공부 하였더라도 핵심이 보이는 간단한 설명을 통해 스케줄링의 특징 기억이 되살아나고 더 나아가 기

역이 공고히 되도록 한다. 그래서 각 스케줄링 환경마다 핵심이 보이는 설명을 추가하였다. 생성된 프로세스의 대기 큐가 여러 개 필요한 정책에서 실습하는 학습자가 대기 큐의 수를 선택을 할 경우 실습자에게 복잡함의 증대를 야기할 수 있다. 따라서 꼭 필요한 경우가 아니면 실습자에게 분심이 일어나지 않게 하여 전달하고자 하는 지식의 핵심을 잃지 않게 단순화를 위해 필요 최소수가 자동 설정되도록 하고 안내 하였다. 수행 도해를 보고 이해할 수 있다고 판단되나, 어떤 시점에 특정 프로세스의 선택 이유에 대한 이해 도움을 위해 스케줄링 정책 특징에 따른 핵심 이유를 추가하였다. 기타 입력 한계를 넘는 행동을 막아준다. 그래서 이런 행동이 발생되면 팝업(pop-up)되는 경고 창을 통해 안내해 주었다.

2. Design

초기화면은 실습을 위한 8개 시뮬레이션 중 1개를 선택하고 선택을 통해 각각의 시뮬레이션을 수행할 수 있다. 시뮬레이션을 수행하는 화면은 공통으로 Fig. 3과 같이 설계하였다. 최상단에는 현재 수행중인 시뮬레이션 명칭을 출력하고, 2번째 창에는 선택한 스케줄링의 특성을 간단히 설명하여 해당 특징 기억이 되살아나게 하였고, 3번째에는 스케줄링 시뮬레이션 자료 입력 창을 구성하되 앞에서 설명한 바와 같이 학습자가 학습을 위해 넣는 것이 좋다고 판단된 것만 최소한으로 입력되도록 하였다. 즉 프로세스 ID 자동 생성 및 선택 스케줄링에 부합하는 입력 환경을 자동 구성되도록 하여 사용자의 편리성을 도모하였다. 입력 구성 참고는 Table. 1과 같다. 4번째 창은 '등록 프로세스 안내 및 스케줄링 도해 설명 창'으로 등록 프로세스 정보를 보여주고 하단 스케줄링 시뮬레이션 도해의 필요 설명을 보충해 주는 내용들이 추가된다. 5번째 창은 '시뮬레이션 시작 또는 종료 버튼 창'으로 시뮬레이션 시작 버튼 / 시뮬레이션 종료 버튼이 있어 설정된 값으로 시뮬레이션을 하거나 종료할 수 있다. 6번째는 '스케줄링 시뮬레이션 도해 창'으로 등록된 프로세스들의 도착 시간, 대기 시간, 응답 시간, 실행 시간, 반환 시간 등을 단위 시간 별로 그림으로 보여주는 창이다. 수강자들의 요구에서 가장 필요로 하는 내용으로 사용자의 시각적 편리를 위해 가장 요구되는 창이다. 7번째는 '통계적 자료 및 프로세스 별 자료 출력 창'으로 시뮬레이션의 목적 중 하나인 선택한 스케줄링의 CPU 이용률, 처리율, 평균 반환 시간, 평균 대기 시간, 평균 응답 시간 및 프로세스 별 응답 시간, 대기 시간, 반환 시간을 볼 수 있도록 구성하였다.

< Process Scheduling Name Output Window >

< Process Scheduling Properties Description Window >
 Review scheduling characteristics through key descriptions of selected scheduling characteristics.

< Scheduling Simulation Data Input Window >
 The window is configured to input only the necessary data for the selected simulation. In addition, the process ID among the input data is automatically generated, and the system calculates the value of the input range for the rest of the data and selects it through the pop-up window to reduce convenience and reduce mistakes of practitioners (including related buttons).

< Registration Process Guidance and Scheduling Diagram Explanation Window >
 Additional information is added that shows the registration process required information and supplements the required description of the scheduling simulation diagram below.

< Simulation Start or End Button Window >
 Simulation Start Button / Simulation End Button

< Scheduling Simulation Illustrated Window >
 It is a window that shows the arrival time, waiting time, response time, execution time, and return time of registered processes by unit time. It is composed for the user's visual convenience with the most necessary contents from the needs of the students.

< Statistical Data Output Window >
 It is configured to view CPU utilization, throughput, average return time, average wait time, average response time and response time for each process, wait time, and return time of the selected scheduling, which is one of the purposes of the simulation.

Fig. 3. Screen Layout

Table 1. Input & Setting Value by Scheduling

scheduling	arrival time	cpu time	priority	cpu allocation time	preemption
FCFS	0	0	X	X	X
SJF	0	0	X	X	X
SRTF	0	0	X	X	0
HRN	0	0	X	X	X
PRI	0	0	0	X	0
RR	0	0	X	0	0
MLQ	0	0	0	X	0
MLFQ	0	0	0	X	0

*. FCFS : First Come First Service, SJF : Shortest Job First, SRTF : Shortest Remain Time First, HRN : Highest Response Ratio Next, PRI : Priority, RR : Round Robin, MLQ : Multi Level Queue, MLFQ : Multi Level Feedback Queue, 0 : Enter or allow, X : Do not enter or disallow

본 시뮬레이션 앱은 다수의 스케줄링 시뮬레이션 수행 시스템이므로 먼저 스케줄링을 선택하고, 선택 시뮬레이션 요구 사항을 입력하고, 프로세스 등록과 시뮬레이션 실행, 시뮬레이션 종료로 요구할 수 있다. 사용자가 할 수 있는 행위는 Fig. 4의 Usecase Diagram에 표시해 놓았다.

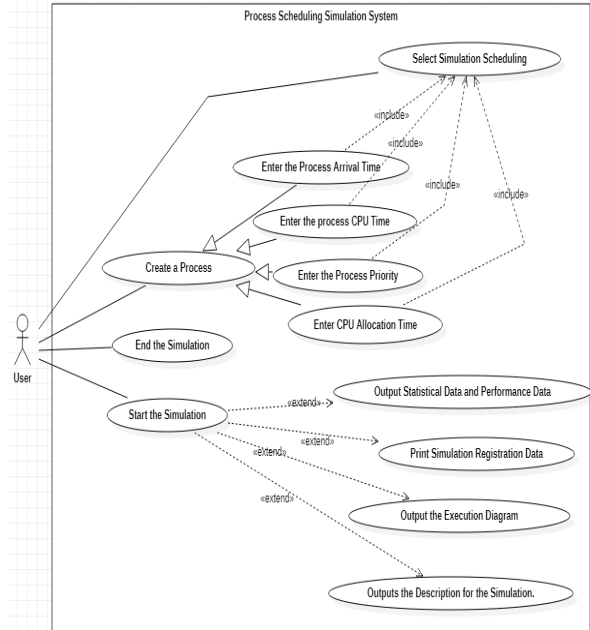


Fig. 4. Usecase Diagram

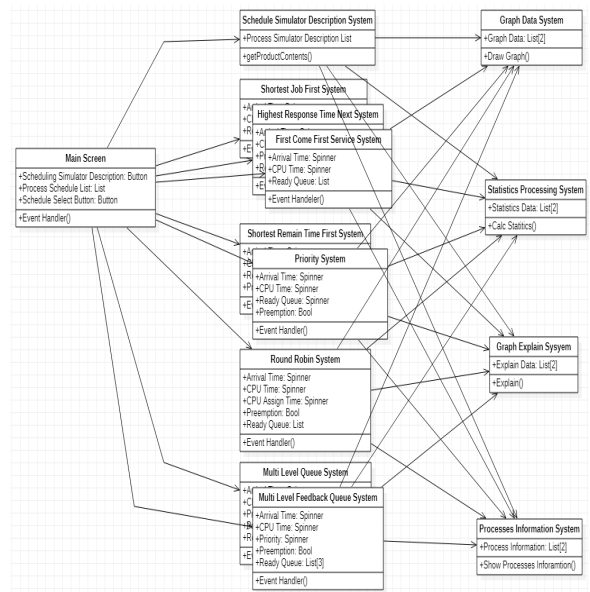


Fig. 5. Class Diagram

프로그램의 수행 과정은 Fig. 5~7을 통하여 학인 할 수 있다. 1차 경계 클래스를 통하여 시뮬레이션 스케줄링을 선택하고, 2차 경계클래스를 통하여 시뮬레이션 요구 사항을 수집하고, 제어클래스를 통하여 프로세스 등록과 시뮬레이션 실행, 시뮬레이션 종료로 요구하거나 다른 시뮬레이션을 위해 전 단계로 되돌아 갈 수 있다. 시뮬레이션 실행은 현재 입력 상태에서 시뮬레이션 실행한 후, 추후 기존 입력에 더하여 프로세스를 추가하고 추가한 상태에서 시뮬레이션을 실행하여 차이를 볼 수도 있다. 시뮬레이션

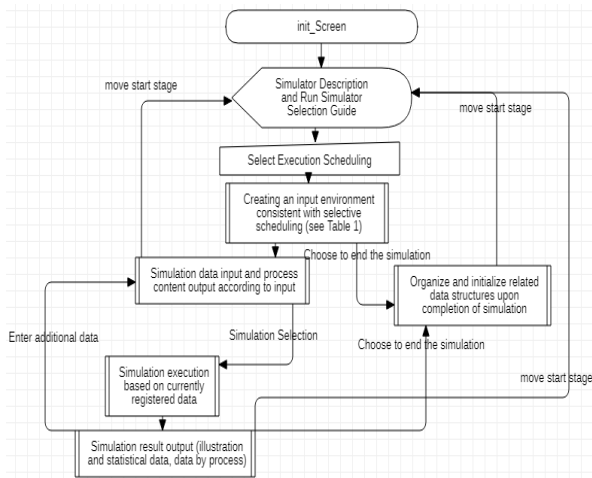


Fig. 6. System Flowchart

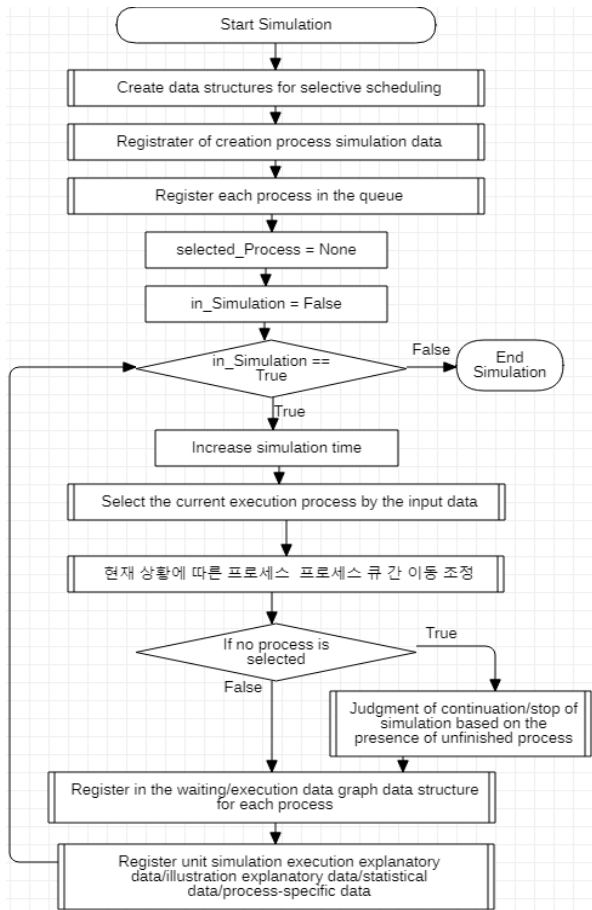


Fig. 7. Simulation Flowchart

종료를 통해 기존에 입력한 프로세스들을 모두 취소하고 처음부터 프로세스들을 모두 다시 입력할 수도 있다. 시뮬레이션 결과는 구성된 Graphic Data System과 Statistics Processing System, Graphic Explain System, Processes Information System의 작동에 의해 볼 수 있다. 그래픽 자료는 행은 단위 시간, 열은 각 프로세스로 구분

되며, 저장되는 값은 'None / Waiting / Running' 으로 시뮬레이션이 수행되면 칸칸이 저장된다. 통계 자료와 설명 자료 그리고 프로세스 별 자료도 해당 이벤트가 발생할 때마다 각각 저장되고, 각 시스템은 이를 이용한다.

IV. Implementation Result and Evaluation

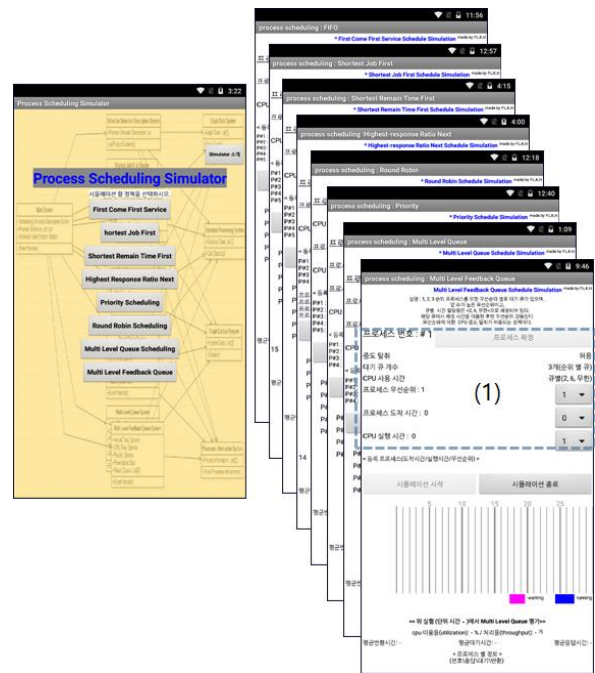


Fig. 8. Implemented Simulator

Fig. 8은 구현된 시뮬레이터의 초기 화면을 통해 각각의 스케줄링 시뮬레이터로 들어갔을 때의 화면이다. 다른 스케줄링은 모두 적당한 실행을 한 상태에서 캡춰되었고, 가장 앞의 Multi Level Feedback Queue 시뮬레이션 화면은 호출 초기 화면이다. 앞에서 설명한 바와 같이 위에서부터 설명하면 첫째로 프로세스 스케줄링 이름 출력되고 있음을 볼 수 있고, 둘째로 프로세스 별로 스케줄링 특성이 설명되고 있음을 볼 수 있다. 셋째는 스케줄링 시뮬레이션 자료 입력 창으로 Fig. 8.에서 (1)로 표현된 영역이다. Table. 1에 표현된 바와 같이 스케줄링 정책에 맞추어 해당 정책에서 필요한 입력이면 합당한 영역의 값을 선택할 수 있도록 스피너가 설정 구현 된다. 그림을 보면 '프로세스 우선순위, 프로세스 도착 시간, cpu 실행 시간'란은 오른쪽에 값을 입력할 수 있는 스피너가 생성되어 있음이 보인다. 아울러 사용자가 스케줄링 정책과 달리 설정하여 실행할 수 있는 것들은 시스템이 내부에서 확정하여 사용

자가 실수하지 않도록 한다. 화면에서는 ‘중도탈취와 대기 큐 개수, CPU 할당 시간’이 고정 값으로 출력되고 있음을 볼 수 있다. 아울러 이 정책에서는 프로세스 우선순위, 프로세스 도착 시간, cpu 실행 시간 값을 스피너를 통해 확정해야 프로세스로 의미를 갖는데 사용자의 해당 동작이 없어 왼쪽의 설정도 무의미한 값이 있음이 보이며, ‘프로세스 확정’ 버튼도 사용할 수 없는 상태(disable)임을 볼 수 있다. 그러나 학습자가 합당한 값을 설정하면 사용할 수 있는 상태(enable)로 전환된다. 넷째 등록 프로세스 안내 및 스케줄링 도해 설명 창은 아직 등록된 프로세스도 없고 시뮬레이션도 수행하지 않아 제목만 보인다. 다섯 번째 시뮬레이션 시작 또는 종료 버튼 창의 시뮬레이션 시작 버튼도 설정된 프로세스가 없어 사용할 수 없는 상태이나, 등록된 프로세스가 있으면 사용 가능 상태로 전환된다. 시뮬레이션 종료 버튼은 언제든지 시뮬레이션을 종료할 수 있게 사용 가능 상태로 설정되어 있다. 여섯번째 스케줄링 시뮬레이션 도해 창은 시뮬레이션을 수행하지 않은 상태라 타임 라인이 설정된 상태와 범례 안내만 보인다. 일곱 번째 통계적 자료 및 프로세스 별 자료 출력 창도 시뮬레이션이 수행되지 않은 상태라 제목만 보인다.

Fig. 9는 Highest Response Ratio Next 정책을 시뮬레이션 한 예이다. 4개의 화면을 모은 것으로 왼쪽과 오른쪽으로 구분하여 상단과 하단은 같은 상태의 화면을 스크롤 하여 관심 부분이 보이도록 한 것이다. 시간적으로 왼쪽이 먼저이고, 오른쪽이 나중이다. 먼저 (도착 시간, CPU 실행 시간)을 (0, 3), (1, 4), (2, 1), (3, 1), (4, 1)의 5개 프로세스를 생성하고 시뮬레이션을 하고, 캡춰한 화면이 왼쪽이며, 이후 (5, 1) 단위 프로세스를 추가 생성한 후 시뮬레이션을 하고, 여기에 또 (6, 1) 단위 프로세스를 추가 생성하고 시뮬레이션을 한 후 캡춰한 화면이 오른쪽이다. A-1 영역을 보면 상단에 위 범례와 같이 생성된 5개의 프로세스 명칭과 함께 각각의 도착 시간과 CPU 실행 시간이 보인다. 하단에는 시뮬레이션 시간과 함께 해당 시간에 의미가 있는 프로세스에 대해 계산된 우선순위 보인다. 이렇게 계산된 우선순위는 C의 도해에 대한 추가 설명으로 이용된다. C 영역을 보면 시뮬레이션 시간 0에 P#1이 실행(running)하고 P#2~5는 아직 존재하지 않는 프로세스라 아무 표시가 없다. 도해의 추가 설명에도 ‘Tm:0 / 우선순위: P#1(1)’로 해당 시간에 존재하는 P#1 프로세스만 우선순위가 (1)로 계산 된 것만 보인다. 시간 1에는 P#1이 수행하고 P#2는 존재하나 대기(waiting)하고 나머지는 존재하지 않아 아무 표시가 없음을 볼 수 있다. 시간 3에는 먼저 도달한 P#2는 대기하고 나중 도착한 P#3 실행하는

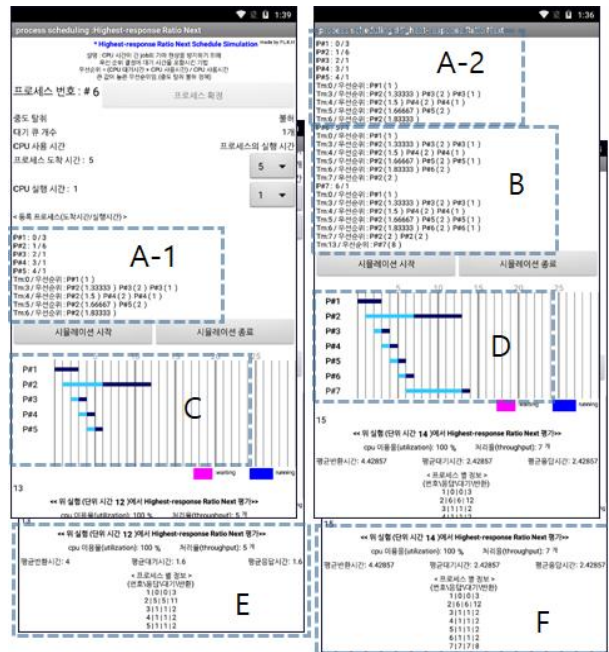


Fig. 9. Examples of Running a Simulation

데 추가 설명을 보면 ‘Tm:3 / 우선순위: P#2(1.33333) P#3(2) P#4(1)’로 출력되어 학습자에게 해당 정책해석에 도움을 주고 있다. 학습자는 계산해 보고 싶을 때 상단의 설명을 참고하여 계산한 후 계산 결과를 비교해 볼 수 있다. 같은 방법으로 B와 D도 해석할 수 있다. 우측의 A-2는 좌측의 A-1과 같아 우측 화면은 한 시뮬레이션에서 자료를 추가하여 확장된 것임을 보여주고 있다. B 영역에 출력을 해석하면 두 번의 확장 시뮬레이션 수행이 있었음을 보여주고 있다. C와 D를 비교하면 P#2가 CPU 실행 시간이 길어 늦게 들어온 짧은 프로세스들에게 계속 우선순위가 밀려 실행되다가 P#7이 아주 짧은 CPU 실행 시간을 갖고 있더라도 이를 제치고 실행되어 기아 상황을 면하게 되는 Highest Response Ratio Next 정책의 특성을 관찰할 수 있게 된다. 이런 상황의 변화는 종이에 인쇄한 교재에서는 구성하기 난해한 것들이다.

E와 F는 각각의 입력에 대해 CPU 이용률, 프로세스 처리율, 평균 반환 시간, 평균 대기 시간, 평균 응답 시간 및 프로세스 별 응답시간, 대기시간 합계, 반환시간을 보여주고 있다. 이런 정보는 학습자가 자신이 이해한 내용을 바탕으로 계산한 것이 맞는지 확인할 수 있는 그래서 학습자가 지식을 확고히 할 수 있는 도움 출력들이다.

Fig. 10은 앞에서 예를 보인 Highest Response Ratio Next를 제외한 나머지 7가지 정책들의 실행 모습과 시뮬레이션을 위해 프로세스의 속성 값을 입력할 때 키보드 없는 스마트 폰 환경을 고려하여 단순한 선택 동작만으로 입

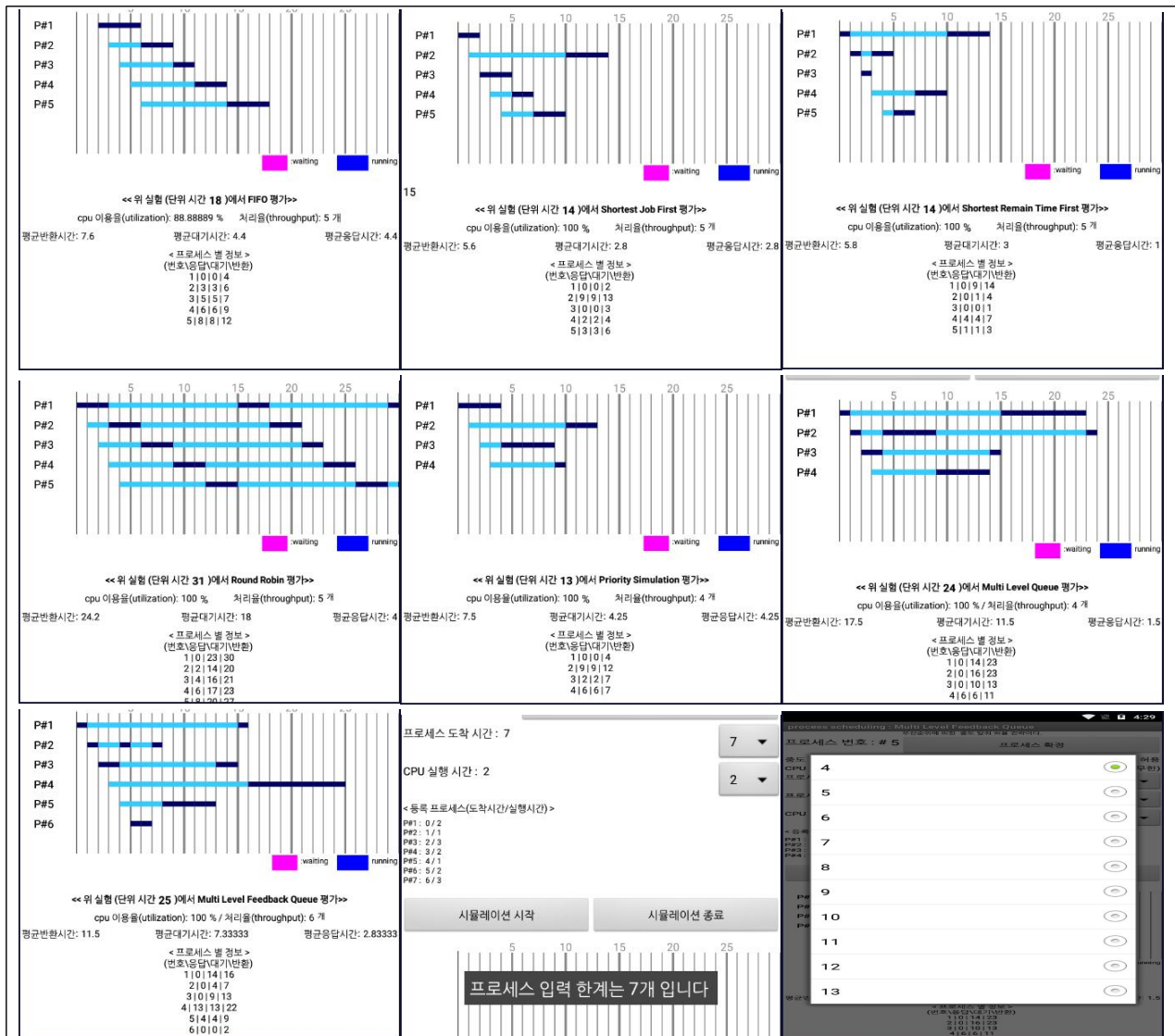


Fig. 10. Examples of Various Simulation

력을 할 수 있게 내부에서 시뮬레이션의 현재 상황을 이용하여 유효한 값을 계산하고 사용자로 하여금 합당한 입력을 할 수 있게 해주는 스피너의 통제된 팝업 값의 출력 모습으로 동일 시간에 2개 이상 발생될 수 없도록 하는 등 사용자가 실수하지 않게 하는 모습과 사용자가 유효 범위를 벗어나는 행위를 하면 경고해 주는 모습을 출력한 것이다.

V. Conclusions

본 연구에서는 학습자들이 다중 프로그래밍의 프로세스 스케줄링 정책 8가지를 잘 이해할 수 있도록 돕는 방법 중 하나로 안드로이드 시뮬레이션 앱을 개발하였다. 학습에서 시뮬레이션 형태의 앱은 실제와 유사한 환경을 제공해야

하며 학습자가 연습할 수 있도록 해야 한다. 이렇게 하기 위해 학습자들의 요구 사항을 조사 분석하고 학습자들의 이해를 중심으로 목적을 정하고, 설계하고 프로그래밍 하여, 8가지 스케줄링 정책을 시뮬레이션 할 수 있는 도구로서의 앱을 만들었고, 이것들이 잘 수행되고 있음을 보였다. 특히 지면 교재의 한계인 단계적 다양한 입력에 대한 단계적 다양한 도해와 그에 따른 설명을 갖지 못하는 문제를 이런 도구를 이용하여 해결할 수 있음을 보여주었다.

아울러 현재 유통되고 있는 디지털 북은 단순히 지면을 디지털화 한 것에 불과한데, 앞에서 보여준 모습을 포함한 형태로 디지털 북을 생산한다면 학습자는 자기 초기 수준에 맞는 단계에서 시작하여 점차 레벨을 올릴 수 있는 단계별 학습이 가능할 것으로 보인다. 향후 많은 다양한 종류의 교육용 시뮬레이션 앱들이 개발되어 학습자들에 큰

도움이 되기 바라며 동시에 디지털 복은 지면 도서의 디지털화로 머무는 것이 아닌 풍부한 디지털 기술이 적용된 도서의 개발이 시작되길 바란다.

[13] Gu, Hyun-He, "Operating System", HanBit Academy, pp.155-168, 2016.

REFERENCES

- [1] Kyung Jin Lee, Kyung Ja Kim, "The Meaning of Curriculum and the Teacher's Role Focusing on the Practical Process of Curriculum Implementation", The Journal of Curriculum Studies, Vol. 23, No. 3, pp. 57-80, 2005 09.
- [2] Hoang Youn-ju, "A Study on the 3D Augmented Reality Animation Education Method for Elementary School Arts Using Smartphone Application 'Quiver'", Korean Elementary Art Education Association, Vol. 60, No. 0, pp. 277-308, 2020 02.
- [3] Ahn Sung-ho, "Research on Teaching Effective Smart-phone Video Production in Film Production Classrooms", ASIAN CINEMA STUDIES, Vol. 10, No. 2, pp. 161-179, 2018 04.
- [4] Eun Joo Yang, Min Sun Kang, "Development of Music-Science Convergence Program(STEAM): Focusing on Popular Music Instrument Making and Smart phone Application Composition", Korean Association of Arts Education, Vol. 13, No. 3, pp. 206-220, 2015 09.
- [5] Kim, Seok-Hun; Woo, Hee-Sun, "Design of Children and Adolescent's Parents for oral health convergence education App in Mobile Environments", Journal of The Korea Society of Computer and Information, Vol. 22 No. 1, pp. 57-62, January 2017.
- [6] JinGu Jung, Consumer Post, <http://www.consumerpost.co.kr/news/articleView.html?idxno=207946>
- [7] Su-jeong Jeong, Keol Lim, Yujung Ko, Hyunae Sim, Kyungyeon Kim, "The Analysis of Trends in Smart Phone Applications for Education and Suggestions for Improved Educational Use", Journal of Digital Contents Society, Vol. 11, No. 2, pp. 203-216, Jun. 2010
- [8] Yeonjung Joe, Seongyoun Hong, "Education Culture Research", Inha University Education Research Institute, Vol. 20, No. 3, pp. 99-124, 2014
- [9] Kyong-ho Lee, "Development of Simulation App for Understanding Test-and-Set Algorithms that Multi Learner Can Use Simultaneously", Journal of The Korea Society of Computer and Information, Vol. 25 No. 9, pp. 0-0, September 2020
- [10] WIKIPEDIA, App Inventor for Android, https://ko.wikipedia.org/wiki/%EC%95%B1_%EC%9D%B8%EB%B2%A4%ED%84%B0
- [11] Choi, Yun-Chul, "Multimedia Course", SengNeungChulPan, pp.28-29, 2014
- [12] Im, SungRag; Jung, Won-il, "Operating System", 21SeGiSa, pp.75-98, 2013

Authors



Kyong-ho Lee received the B.S. degree in Computer Science from Korea National Open University and the M.S. degree in Information and Communication Engineering from Korea Advanced Institute of Science

and Technology and the Ph.D. degrees in Electronic Engineering from Dankook University, Korea, in 1991, 1994 and 2008, respectively. Dr. Lee is currently a Professor in the School of Information & Communication, Broadcasting Engineering, Halla University. He is interested in HCI.