**소프트웨어 공학**

**유민수 교수님**

**컴퓨터공학부 컴퓨터전공**

**2013011424**

**성예닮**

**Project 1**

**설계 문서**

**(Design Document)**

다음은 UML 사용 케이스 다이어그램이다.



사용법은 아주 간단하며, ­각각의 과정은 모두 버튼으로 되어있다.

스케줄링 버튼은 입력 받은 파일에 대하여 스케줄링을 하는 과정이다.

과정 출력 버튼은 스케줄링 과정을 출력한다.

결과 출력 버튼은 최종적으로 스케줄링 한 결과 (처리 된 작업, 처리되지 않은 작업 등)를 출력한다.

프로그램 종료 버튼은 언제든지 프로그램을 종료하는 기능이 있다.

Software Project 버튼은 클릭 시 시뮬레이터의 이름과, 제작자, 용도 등을 보여준다

다음은 UML 클래스 다이어그램이다.









Input file은 텍스트 형식으로 된, 주어진 형식에 맞는 입력 파일이다.

Reader 클래스는 입력 파일을 읽어서 작업 발생기에게 넘겨주는 역할을 한다.

Task Generator는 Reader가 읽은 파일에 대한 정보들을 받아 작업을 생성한다.

만들어진 작업들은 Task라는 객체들의 배열인 Task Array에 저장된다.

Task 객체는

작업 이름, 작업 도착 시각, 작업 기간, 작업 실행 시각, 마감 한계, 인스턴스 이름, 작업 시작 시각, 블로킹 된 시간, 작업 재개 시간, 작업 종료 시간, 작업 상태 (처리 되었는지, 되지 않았는지), 작업 형태 (주기적인지, 비주기적인지)를 나타내는 변수를 갖는다.

시뮬레이터에서는 메인 함수와 작업 발생기, 모드 선택을 할 수 있다.

모드는 1, 2, 3 세 가지이고, 각각은 RM Mode, EDF Mode, User Mode로 구분되며, 해당하는 알고리즘에 맞춰 스케줄링 하고 과정을 처리하는 역할을 한다.

Writer 클래스는 각각의 모드로부터 처리된 결과물을 받아, Console. GUI 환경에 시각적으로 디스플레이 하고 형식에 맞는 출력 파일을 생성한다.

다음은 순서 다이어그램이다.

사용자가 Reader에게 입력 파일을 넘겨준다.

Reader가 사용자로부터 넘겨받은 파일로부터 작업들을 생성하여 Simulator에게 넘겨준다.

Simulator가 Reader로부터 넘겨받은 작업들을 스케줄링하고 Writer에게 넘겨준다.

Writer는 넘겨받은 스케줄링 결과를 콘솔 혹은 GUI 환경을 통해 사용자에게 보여준다.

다음은 상태 표 다이어그램이다.



프로그램을 실행한다.

메인 화면을 보며 원하는 행동을 한다.

입력 파일을 입력한다.

스케줄링 알고리즘을 선택한다.

사용자 지정 스케줄링 알고리즘을 선택할 경우, 지정 방식대로 스케줄링을 할 수 있다.

프로그램이 출력 파일을 출력하고, 사용자가 올바른지 확인한다.

프로그램을 종료한다.

다음은 콜라보레이션 다이어그램이다.



프로그램을 실행한다.

형식에 맞는 입력 파일을 만들고, 입력 파일을 입력한다.

RM, EDF, 사용자 우선순위 중 원하는 방식을 선택하고 스케줄링을 실행한다.

스케줄링 한 결과를 파일 혹은 프로그램이 저장하고 있다.

콘솔 혹은 GUI환경으로 사용자에게 출력한다.

프로그램을 종료하지 않고, 다시 한번 스케줄링 하려면 입력 파일을 새로 입력하면 된다.