
운영체제 과제1 보고서



과목: 운영체제 3분반
교수명: 김하연 교수님
학과: 모바일시스템공학과
학번: 32224020
이름: 전은지

Index

I. Introduction

II. Body

1. Background information
2. Program description
3. Result

III. Conclusion

I. Introduction

이번 보고서는 프로세스 스케줄링과 멀티 태스킹의 기본을 시뮬레이션하기 위해 개발된 파이썬 기반 시뮬레이터에 관한 것이다.

이번 과제는 Round Robin 스케줄링 알고리즘을 사용하여, 여러 프로세스가 시스템 자원을 공유하며 동시에 실행될 수 있는 환경을 시뮬레이션함으로써, 프로세스 스케줄링의 기본 원리를 실습하고 이해하는 것이 목적이다. 또한, 이 시뮬레이터는 시각적 도구를 통해 프로세스의 상태 변화와 스케줄링 알고리즘의 동작을 직관적으로 보여주는 기능을 포함하고 있다.

II. Body

1. Background information

1) CPU 스케줄링

CPU 스케줄링이란 준비 큐(ready queue)에 있는 프로세스 중 어떤 것에 CPU를 할당할지 결정하는 과정으로 프로세스의 실행 순서를 결정하여 CPU 이용률, 처리량, 응답 시간 등을 최적화하는 스케줄링을 말한다. CPU 스케줄링은 프로세스가 실행 상태에서 대기 상태로 전환될 때, 실행 상태에서 준비 상태로 전환될 때, 대기 상태에서 준비 상태로 전환될 때, 프로세스가 종료할 때 발생할 수 있으며 프로세스가 실행 상태에서 대기 상태로 전환될 때와 프로세스가 종료할 때는 필수적으로 발생한다. CPU 스케줄링의 주요 평가 기준은 CPU 이용률, 처리량, 총 처리 시간, 응답 시간, 대기 시간이다.

2) 선점형 스케줄링

선점형 스케줄링이란 운영 체제가 프로세서 자원을 선점하고 있다가 각 프로세스의 요청이 있을 때 특정 요건들을 기준으로 자원을 배분하는 방식이다. 현재 프로세스보다 더 짧은 잔여 시간을 가진 프로세스가 준비 큐에 도착하면, 현재 프로세스는 중단되고 새 프로세스가 CPU를 할당 받는 SRT 방식을 따른다. 프로세스가 실행 중이면 새로 도착하는 프로세스의 잔여 실행 시간과 현재 실행 중인 프로세스의 남은 시간을 비교하여 처리한다. 이 스케줄링은 실시간으로 시스템 상태를 모니터링하고, 새로운 프로세스의 도착에 따라 신속하게 대응할 수 있다.

3) 비선점형 스케줄링

비선점형 스케줄링은 어떤 프로세스가 CPU를 할당 받으면 그 프로세스가 종료되거나 입출력 요구가 발생하여 자발적으로 중지될 때까지 계속 실행되도록 보장하는 방식이다. 준비 큐에 있는 프로세스 중 예상 CPU 버스트 시간이 가장 짧은 것부터 우선적으로 CPU를 할당 받는다. 이 스케줄링은 이론적으로 최단 작업을 먼저 처리함으로써 평균 대기 시간을 최소화할 수 있고, 프로세스의 다음 CPU 버스트 시간을 예측하기 위해 과거의 버스트 시간의 평균을 계산하는 간단한 예측 모델이다.

4) 라운드 로빈 스케줄링

라운드 로빈 스케줄링은 시분할 시스템을 위해 설계된 선점형 스케줄링의 하나로서, 프로세스들 사이에 우선순위를 두지 않고, 순서대로 시간단위(Time Quantum)로 CPU를 할당하는 방

식의 CPU 스케줄링 알고리즘이다. 각 프로세스는 CPU를 사용할 수 있는 고정된 시간을 FIFO 순서로 할당 받는다. 각 프로세스에 할당 시간이 끝나면 타이머 인터럽트가 발생하고, CPU는 다음 프로세스로 전환된다. 이 스케줄링은 모든 프로세스가 동일한 시간동안 CPU를 사용할 기회를 얻기 때문에 공평하다. 또한, 이 방식을 사용하면 짧은 작업이나 대화형 작업은 긴 작업에 의해 지연되지 않으므로 시스템의 전반적인 응답 시간이 단축된다는 장점이 있다. 시간 할당량이 명확하게 정의되어 있기 때문에, 시스템의 동작이 예측 가능하고 관리하기 쉬운 스케줄링 방식이다.

2. Program description

1) 프로세스 클래스

'Process' 클래스를 선언한다. 해당 클래스는 각 프로세스의 ID, 버스트 타임(burst time), 남은 실행 시간(remaining time) 및 현재 상태(state)를 포함하여 프로세스의 주요 특성을 정의한다. 프로세스의 상태는 초기에 'Ready'로 설정되며, 실행 중에는 'Running'으로 변경되고, 실행이 완료되면 'Completed'로 업데이트 된다.

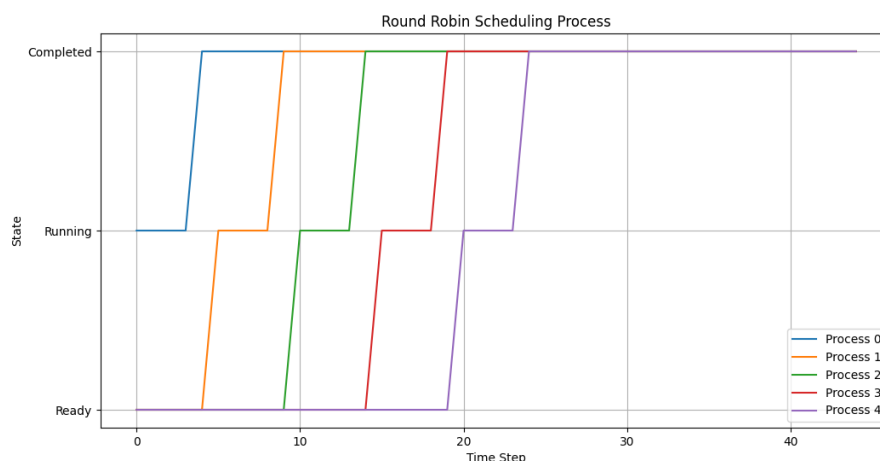
2) 라운드로빈 스케줄러 클래스

'RoundRobinScheduler' 클래스를 선언한다. 이 스케줄러는 프로세스 큐를 관리하고 타임 쿼텀을 기반으로 각 프로세스에 CPU 시간을 할당하며 프로세스가 완료되면 다음 프로세스를 큐에서 꺼내 실행 상태로 전환한다.

3) 메인 실행 루프

먼저, 5개의 프로세스와 타임 쿼텀을 1로 설정한 라운드로빈 스케줄러를 생성한 후 모든 프로세스를 스케줄러 큐에 추가한다. 스케줄러 큐가 비어있지 않다면 계속해서 스케줄링이 진행된다. 현재 프로세스는 None이 아닐 경우 스케줄링이 진행되고 자신의 'remaining_time'을 감소시켜 0이 되면 'Completed'로 상태를 변경한다. 프로세스 진행 과정을 시각화하기 위하여 각 타임 스텝마다 프로세스 상태를 추적하고 해당 데이터를 이용하여 시각화한다.

3. Result



III. Conclusion

이번 과제를 통해 파이썬을 이용하여 Round Robin 스케줄링 알고리즘을 구현함으로써 다수의 프로세스가 공평하게 CPU 시간을 할당 받는 환경을 모델링하였다. 이 과정에서 각 프로세스의 상태 변화를 시각적으로 표현하는 기능을 추가하여, 시뮬레이션의 동작을 직관적으로 이해할 수 있도록 하였다. 실행 결과를 시각화 함으로써 하나의 프로세스가 완료되면 다른 프로세스로 넘어가는 과정과 각 프로세스가 공평하게 시간을 할당 받은 사실을 직접 확인할 수 있었고 스케줄링을 이해하는데 도움이 되었다. 과제는 스케줄링의 기본 원리와 운영체제의 멀티태스킹 기능이 어떻게 구현 될 수 있는지에 대해 생각해보게 하였고 이를 통해 더 다양한 스케줄링 알고리즘에 대한 흥미를 얻게 되었다.