

Project: Process Scheduling

Multi-level Feedback Q Scheduler

하나의 프로세스는 살아있는 동안 여러 개의 큐를 옮겨 다닌다. 운영체제는 각 큐에서 적절한 프로세스를 선택한다. 이 선택하는 과정은 스케줄러에 의해서 결정된다.

Multi-level Scheduling 알고리즘에서는 프로세스들은 시스템에 들어오는 순간부터 끊임없이 하나의 큐에 할당된다. 프로세스들은 큐 사이를 돌아다니지는 않는다. 하지만 Multi-level Feedback queue scheduling에서는 큐 사이의 이동을 허용한다. 아이디어는 프로세스를 다른 CPU-burst 특성으로 분리하는 것이다.

일반적으로 Multi-level Feedback Q Scheduler는 다음과 같은 특성으로 정의된다.

1. 큐의 수
2. 각 큐의 스케줄링 알고리즘
3. 하나의 프로세스를 우선권이 높은 큐로 이동시킬 때 결정하는 방법
4. 하나의 프로세스를 우선권이 낮은 큐로 이동시킬 때 결정하는 방법
5. 하나의 프로세스가 서비스를 받으러 들어왔을 때 어떤 큐를 선택할지 결정하는 방법

위의 정의는 가장 일반적인 스케줄링 알고리즘이다. Multi-level Feedback Q Scheduler는 가장 일반적인 스케줄링 알고리즘이지만 가장 복잡한 알고리즘이기도 하다.

요구사항 & 파라미터

Multi-level Feedback Q Scheduler는 다음과 같은 파라미터를 만족해야 한다.

- 스케줄러는 4개의 일직선 Q로 이루어져 있다.
- 첫 큐는 FIFO(First-In First-Out) 알고리즘, 두 번째는 우선순위 Q 알고리즘, 세 번째 Q는 SJF(Shortest Job First) 알고리즘, 네 번째는 가장 우선순위가 높는데, Round-Robin 알고리즘을 사용한다.
- Feed back은 aging을 사용한다. 그러나 각 큐는 보다 높은 우선순위를 가진 큐로 이동하기 위한 다른 aging threshold(시간)를 가진다.
- Round Robin 큐의 Quantum은 변수 파라미터이다.
- SJF, 우선순위 Q에서 preemptive인지 non-preemptive인 입력할 수 있는 변수를 가져야 한다.
- 모든 상위 우선순위를 가진 큐에 프로세스가 없을 경우에만 아래의 큐에서 프로세스가 실행될 수 있다.
- PCB는 동적이다. 즉, scheduler가 동작하고 있을 동안에 프로세스가 도착한다.
- 동작마다 출력이 화면에 되어야 한다. 즉, 어떤 프로세스가 생성되었는지 어떤 프로세스가 동작을 마쳤는지, 어떤 프로세스가 다른 큐로 이동했는지를 표시해야 한다.
- Context switching time은 다르다.
- 각 큐의 형식에 따라 예측 시간이 다르다.

구현된 프로그램은 다변량 최적화 프로그램과 같이 여러 입력을 사용하여 동작해야 한다.

프로그램을 설계할 때, 다음과 같은 파라미터를 사용해야 한다.

- Context
- 3개의 aging 파라미터
- Time slot
- Preemption/Non-preemption을 위한 2개의 스위치
- 4개의 예측 계수

파라미터 입력에
첫 몇 단계 출력

Enter the time cycles : 100

Enter the value of time slot for RR : 1

Enter the context switching time : 1

SJF with pre-emption (1-yes/0-no) : 1

PQ with pre-emption (1-yes/0-no) : 0

Enter the alpha co-eff for RR : 1

Enter the alpha co-eff for SJF : 2

Enter the alpha co-eff for PQ : 3

Enter the alpha co-eff for FIFO : 3

Enter the aging time for RR : 9

Enter the aging time for SJF : 5

Enter the aging time for PQ : 9

Enter the aging time for FIFO : 5

Enter the initial estimated time for RR : 5

Enter the initial estimated time for SJF : 9

Enter the initial estimated time for PQ : 4

Enter the initial estimated time for FIFO : 7

Process appended to SJF queue.
Process execution time = 9

of jobs in RR = 0

of jobs in SJF = 1

of jobs in PRQ = 0

of jobs in FIFO = 0

#context switches = 1

#Total processes generated = 1

#Total processes completed = 0

no new process was generated

of jobs in RR = 0

of jobs in SJF= 1

of jobs in PRQ = 0

of jobs in FIFO = 0

#context switches = 1

#Total processes generated = 1

#Total processes completed = 0

no new process was generated

of jobs in RR = 0

of jobs in SJF= 1

of jobs in PRQ = 0

of jobs in FIFO = 0

#context switches = 1

#Total processes generated = 1

#Total processes completed = 0

Process appended to PRQ queue.

Process execution time = 4

of jobs in RR = 0

of jobs in SJF= 1

마지막 몇 단계

of jobs in PRQ = 1

of jobs in FIFO = 0

#context switches = 1

#Total processes generated = 2

#Total processes completed

Process appended to SJF queue.

Process execution time = 2625

of jobs in RR = 22

of jobs in SJF= 9

of jobs in PRQ = 9

of jobs in FIFO = 2

#context switches = 44

#Total processes generated = 45

#Total processes completed = 3

SUMMARY

AVERAGE WAITING TIME = 30.3333 time units

AVERAGE TURNAROUND TIME = 34.3333 time
units

CPU UTILIZATION = 100 %

MAXIMUM TURNAROUND TIME = 59 time units

MAXIMUM WAIT TIME = 54 time units

CPU THROUGHPUT = 0 %

성능 측정 기준

목적은 다음과 같은 성능 기준에 최적화 하는 것이다.

- Average waiting time
- Average turnaround time
- CPU utilization
- Maximum turnaround time
- Maximum waiting time
- CPU throughput