

1. [5] X 시스템의 현재 Ready Queue 에 포함된 프로세스의 Arrival Time 과 CPU Burst 시간이 다음과 같다고 가정하자. 이 때, FCFS, Nonpreemptive SJF, 및 RR 스케줄링 알고리즘을 사용했을 때 각 프로세스의 Waiting Time(WT)과 Turnaround Time(TRT)를 계산하시오. 단, Time Quantum 의 크기는 1 이며, Context Switch 에 소비되는 시간은 0 으로 가정한다. 모든 조건이 동일한 경우 먼저 도착한 프로세스를 먼저 처리하도록 한다.

Process ID	Arrival Time	CPU Burst Time
Process-1	0	4
Process-2	2	5
Process-3	3	3
Process-4	8	4

Scheduler	Time	Process-1	Process-2	Process-3	Process-4
FCFS	WT	0	2	6	4
	TRT	4	7	9	8
Nonpreemptive SJF	WT	0	5	1	4
	TRT	4	10	4	8
RR	WT	4	8	5	4
	TRT	8	13	8	8

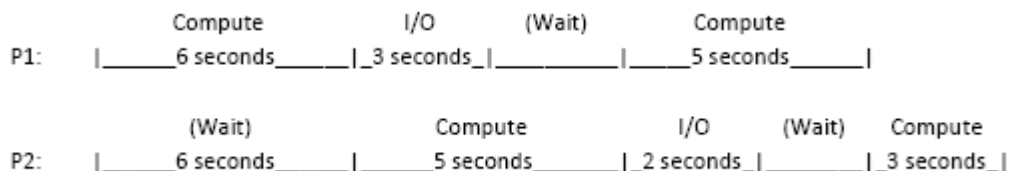
2. [5] 시스템 X 는 FCFS 스케줄링 알고리즘이 적용된 Multiprogramming 시스템이다. 현재 시스템 X의 Ready Queue 에는 프로세스 P1 과 P2 가 있으며 각 프로세스의 실행 내용 및 순서는 다음과 같다.

P1: Computation (6 초) → I/O (3 초) → Computation (5 초)

P2: Computation (5 초) → I/O (2 초) → Computation (3 초)

P1, P2 프로세스의 도착 시간은 둘 다 0 이며, P1 에게 CPU 가 먼저 할당된다고 가정했을 때 각 프로세스의 간트 차트를 그리고 Ready Queue 에서 보내는 시간을 계산하시오.

[Answer]



P1: 2 초, P2: 9 초

3. [5] Multi-Threading 기법을 이용하여 Video Streaming 을 위한 YouTube 웹서버를 개발하고자 한다. 사용자가 특정 비디오 파일을 요청하면, 해당 요청을 담당하는 스레드는 디스크로부터 파일을 읽어들이 스트리밍을 수행한다. 이 때 디스크 I/O 를 위해 운영체제는 Blocking read 만을 위한 시스템 콜을 제공한다고 가정했을 때 User-Level Thread 와 Kernel-Level Thread 중 어떤 형태의 스레드를 이용하여 구현하는 것이 유리한지 상세히 기술하시오.

[Answer]

Kernel-Level Thread. 각각의 Thread 가 Blocking I/O 를 호출할 수 있기 때문에 User-Thread 의 경우 한 스레드가 호출하면 다른 스레드들이 모두 블록되어야 하지만 Kernel Thread 의 경우 그렇지 않음

4. [5] Multi-level feedback queue 스케줄링 알고리즘은 IO-bound process 를 선호하여 입출력 장치의 활용도를 높인다고 볼 수 있는가? Yes 또는 No 로 답하고 그 원인을 상세히 기술하시오.

[Answer]

IO-bound process 는 CPU 사용시간이 짧다. 따라서 multi-level feedback queue 에서 우선 순위가 높은 queue 에서 우선 실행되기 때문에 IO 활용도를 높일 수 있다.

5. [21] 스케줄링 알고리즘 X 가 있다. X 알고리즘은 시스템에 새롭게 진입한 프로세스들을 위한 queue (NewQ)과 기존의 프로세스들을 위한 queue (OldQ)로 구성되면 동작 과정은 다음과 같다.

- ① 새로운 프로세스가 시스템에 진입하게 되면 NewQ 의 끝에 삽입된다.
- ② NewQ 에서 2 msec 경과 후 OldQ 로 옮겨진다
- ③ 새로운 프로세스에게 CPU 를 할당하게 될 경우 X 알고리즘은 두 개의 queue 를 번갈아가면 각 queue 의 가장 먼저 저장된 프로세스를 선택하게 된다. (예를 들어, NewQ 에 가장 먼저 저장된 프로세스에게 CPU 를 할당하고, 그 다음 번에는 OldQ 에서 가장 먼저 저장된 프로세스에게 CPU 를 할당)
- ④ X 알고리즘은 nonpreemptive 로 동작
- ⑤ 프로세스들은 임의의 시간에 시스템에 진입할 수 있으며, 대부분의 프로세스들은 2msec 보다 훨씬 많은 시간을 필요로 한다고 가정한다. 단 프로세스들은 무한루프와 같은 비정상 동작은 하지 않는다고 가정한다.

- 1) X 알고리즘은 신규 프로세스에게 가장 높은 우선순위를 부여하는가? Yes 또는 No 로 대답하고 그 이유를 상세히 기술하시오.

[Answer]

No.

첫째, X 알고리즘은 NewQ와 OldQ를 번갈아가면서 다음 CPU를 할당받을 프로세스를 선택하기 때문에 신규 프로세스가 항상 최상위 우선순위를 가지지 못한다.

둘째, 현재 실행 중인 프로세스가 2msec 이상 소요할 경우 새롭게 도착한 프로세스는 OldQ의 가장 뒤에 위치하기 때문에 OldQ의 가장 앞에 위치한 프로세스보다 우선 순위가 뒤쳐지게 된다.

- 2) X 알고리즘을 사용할 경우 starvation 문제가 발생하는가? Yes 또는 No로 대답하고 그 이유를 상세히 기술하시오.

[Answer]

No.

신규 프로세스는 20msec 이후에 OldQ로 이동하게 되며, 시간이 지남에 따라 점점 앞으로 이동하기 때문에 궁극적으로는 OldQ의 맨 앞에 위치하게 됨. 따라서 starvation 문제가 발생하지 않음

- 3) X 알고리즘은 공평한가? 즉, 모든 프로세스들이 평균적으로 유사한 waiting time을 가지는가? Yes 또는 No로 대답하고 그 이유를 상세히 기술하시오.

[Answer]

No

신규 프로세스가 시스템에 진입하여 NewQ에 들어가자마자 바로 CPU를 할당받는 경우도 발생할 수 있고, OldQ로 이동하여 신규 프로세스 이전에 시스템에 진입하여 OldQ로 이동한 프로세스들이 모두 실행이 끝날 때까지 기다려야 하는 경우도 발생하기 때문에 공평하지 않음.

6. [14] Lottery 스케줄링 알고리즘은 각 프로세스에게 복권(Lottery Ticket)을 발급하며, 무작위로 복권을 선택하여 해당 복권을 보유한 프로세스에게 CPU를 할당한다. 이 때 복권의 개수를 차등 지급하여 우선 순위가 높은 프로세스는 우선 순위가 낮은 프로세스보다 더 높은 당첨 가능성을 확보하도록 한다. Lottery 스케줄링 알고리즘은 기존의 단순히 우선 순위를 비교하여 우선 순위가 높은 프로세스에게 CPU를 할당하는 우선순위 스케줄링 알고리즘에 비해 어떠한 장점을 가지고 있는지 기술하시오. 만약 특별한 장점이 없다면 "없음"으로 작성하시오.

[Answer]

기존의 우선순위 스케줄링 알고리즘에서는 우선 순위가 높은 프로세스가 계속 진입할 경우 Starvation 문제가 발생한다. 그러나 Lottery 스케줄링 알고리즘에서는 우선 순위가 높은 프로세스가 더 많은 복권을 보유하고 있으나 반드시 그 복권이 선택된다는 가정은 할 수 없기 때문에 기존 우선순위 스케줄링 알고리즘에 비해 starvation이 발생할 확률이 상대적으로 떨어진다.

7. [10] CPU Scheduler 는 프로세스의 특성 (IO-bound 프로세스와 CPU-bound 프로세스)을 고려하여 CPU 를 차등적으로 할당하는 것이 중요하다. 그러한 이유를 설명하시오.

[Answer]

IO-bound 프로세스는 상대적으로 CPU burst 가 CPU-bound 프로세스보다 짧다. 따라서, CPU Burst 가 짧은 프로세스에게 CPU 를 할당함으로써 전체 프로세스의 대기 시간을 줄일 수 있다