

스케줄링

유튜브 주소: <https://www.youtube.com/watch?v=H4tmie2IOqE>

스케줄링의 이해

스케줄링의 단계

스케줄링 알고리즘

스케줄링의 개념

- 스케줄링(Scheduling)의 개념

- 여러 프로세스가 번갈아 사용하는 자원을 어떤 시점에 어떤 프로세스에 할 당할지 결정.

- 자원이 프로세서인 경우를 프로세서 스케줄링, 대부분의 스케줄링이 프로세서 스케줄링 의미

- 스케줄링 방법에 따라 프로세서를 할당받을 프로세스 결정하므로 스케줄링이 시스템의 성능에 영향 미침

- 좋은 스케줄링은 프로세서 효율성 높이고, 작업(프로세스)의 응답시간 최소화하여 시스템의 작업 처리 능력 향상

스케줄링의 목적

- 자원 할당의 공정성 보장
 - 적절한 반환시간 보장
 - 오버헤드 최소화
 - 반환시간과 자원의 활용 간에 균형 유지
- 단위 시간당 처리량 최대화
 - 예측 가능성 보장
 - 자원 사용의 균형 유지
 - 실행의 무기한 연기 방지
우선순위

스케줄링의 단계

• 스케줄링 수행 단계

1단계 작업 스케줄링: 작업 선택

실제로 시스템 자원을 사용할 작업 결정하는 작업 스케줄링. 승인 스케줄링이라고도 함
작업 스케줄링에 따라 작업 프로세스들로 나눠 생성. 수행 빈도가 적어 **장기 스케줄링**에 해당

2단계 작업 승인과 프로세서 결정 스케줄링: 사용 권한 부여

프로세서 사용 권한 부여할 프로세스 결정하는 작업 승인과 프로세서 할당 스케줄링

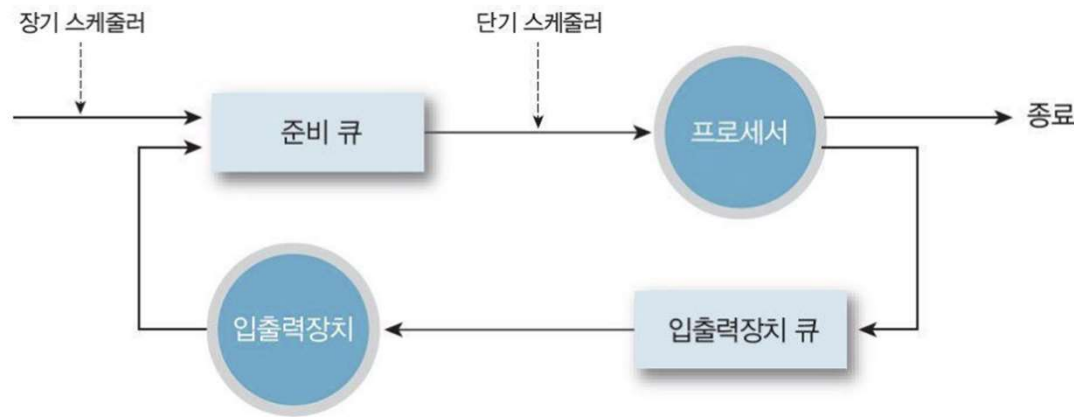
시스템의 오버헤드에 따라 연기할 프로세스 잠정적으로 결정. 1단계 작업 스케줄링과 3 단계 프로세서 할당 스케줄링의 완충 역할. 수행 빈도를 기준으로 하면 **중기 스케줄링**에 해당, **메모리 사용성도 높이고 작업 효율성 향상시키는 스와핑swapping(교체) 기능의 일부로 이해 가능**

3단계 프로세서 할당 스케줄링: 준비 상태의 프로세스에 프로세서 할당(디스패칭)

디스패처(분배기)가 준비 상태에 있는 프로세스 중에서 프로세서를 할당할 프로세스를 결정하는 스케줄링.
단기 스케줄링에 해당

스케줄링과 스케줄러

• 스케줄러의 종류와 역할



장기 스케줄러

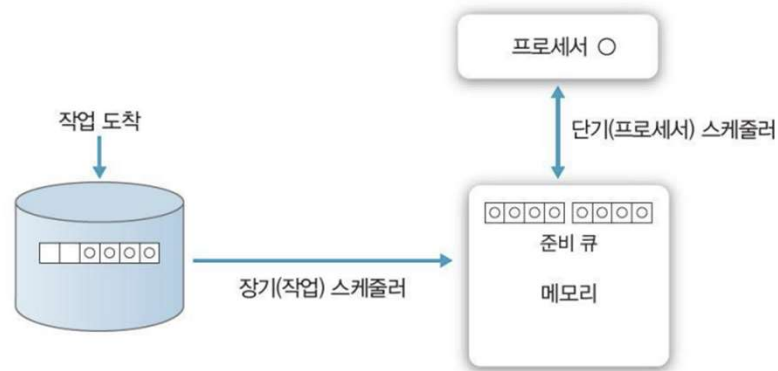
작업 스케줄러라고도 하며, 스케줄링에 따라 디스크에서 메모리로 작업 가져와 처리할 순서 결정. 제출 시간, 작업 이름, 작업 길이(용량) 등의 정보 필요

단기 스케줄러

메모리에 적재된 프로세스 중 프로세서를 할당하여 실행 상태가 되도록 결정하는 프로세스 스케줄링을 한다. 이때는 프로세스가 실행하는 데 필요한 자원의 요청 만족해야 함

스케줄링의 단계

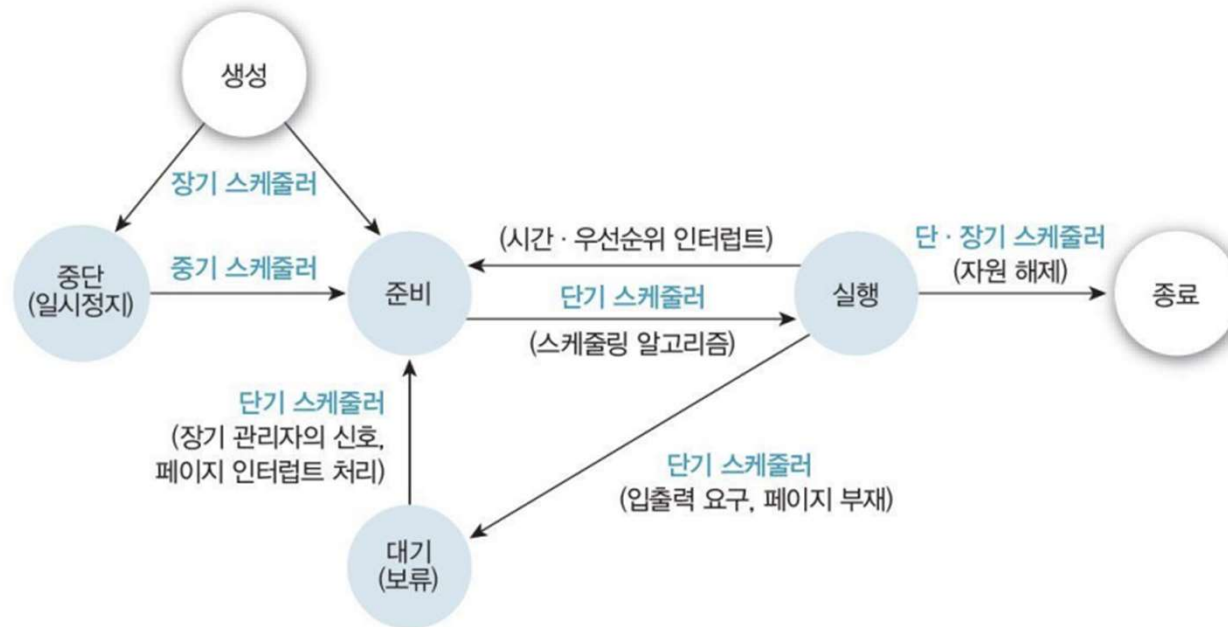
- 장기 스케줄러와 단기 스케줄러 차이



- 장기 스케줄러와 단기 스케줄러의 가장 큰 차이는 실행 빈도
- 단기 스케줄러는 실행할 프로세스 수시로 선택, 매우 빨라야 함
- 장기 스케줄러는 시스템에 새로운 작업이 분minute 단위로 들어오므로 단기 스케줄러에 비해 상대적으로 드물게 실행.
- 장기 스케줄러는 다중 프로그래밍의 정도degree of multiprogramming(메모리에 있는 프로세스 수) 결정

스케줄링 단계

- 프로세스 상태 변화와 스케줄러의 역할



스케줄링 알고리즘

- 선점 스케줄링

프로세스 하나가 장시간 동안 프로세서 독점 방지하여 모든 프로세스에 프로세서를 서비스할 기회 늘림.

따라서 우선순위가 높은 프로세스들이 긴급 처리 요청할 때 유용

실시간 시스템에서 인터럽트를 받아들이지 않으면 결과는 예측 불가

대화식 시분할 시스템이나 실시간 시스템에서 빠른 응답시간 유지 위해 필수

선점 동작에 오버헤드 있음 되도록 메모리에 프로세스가 많이 적재될 필요

프로세서를 사용 가능할 때마다 실행할 수 있는 프로세스들이 준비 상태에 있어야 효과적

우선순위라는 개념을 반드시 고려, 우선순위는 의미 있게 부여하지 않으면 효과 없음

스케줄링 알고리즘

• 비전섬 스케줄링

실행 시간이 짧은 프로세스(작업)가 실행 시간이 긴 프로세스(작업)를 기다리는 대신 모든 프로세서를 공정하게 관리

우선순위가 높은 프로세스 중간에 입력해도 대기 중인 프로세스는 영향을 받지 않으므로 응답시간 예측 용이

스케줄링 알고리즘 비교 시 참조 기준

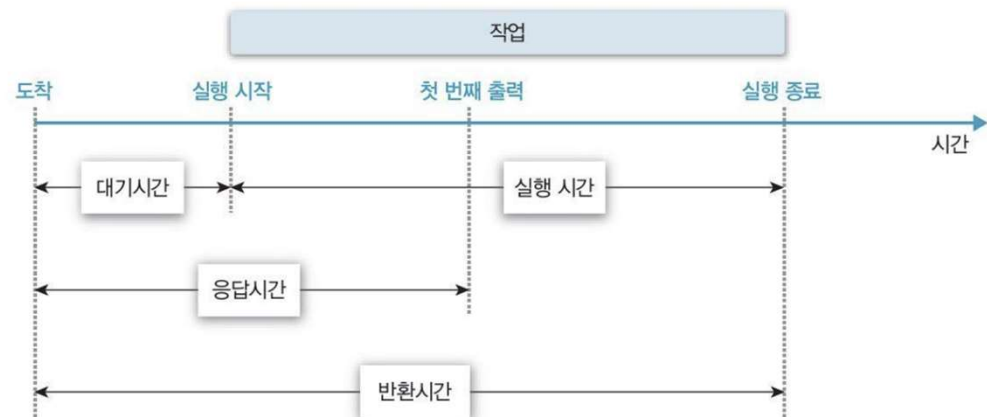
프로세서 사용률

처리율

반환시간

대기시간

반응시간



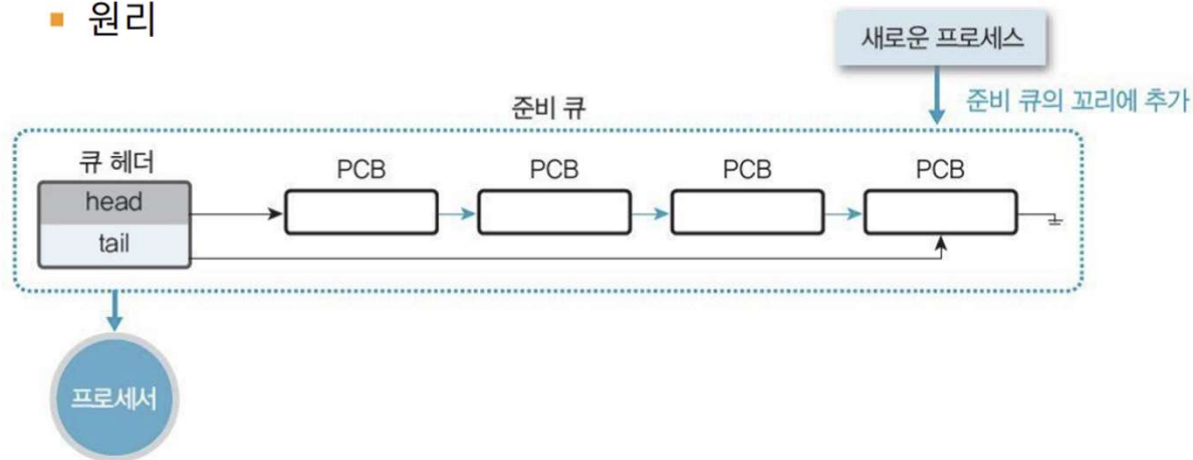
스케줄링 알고리즘

- 선입선처리 스케줄링의 개념

비선점 방법으로 프로세서 스케줄링 알고리즘 중 가장 단순
프로세서 요청하는 순서대로 프로세서 할당

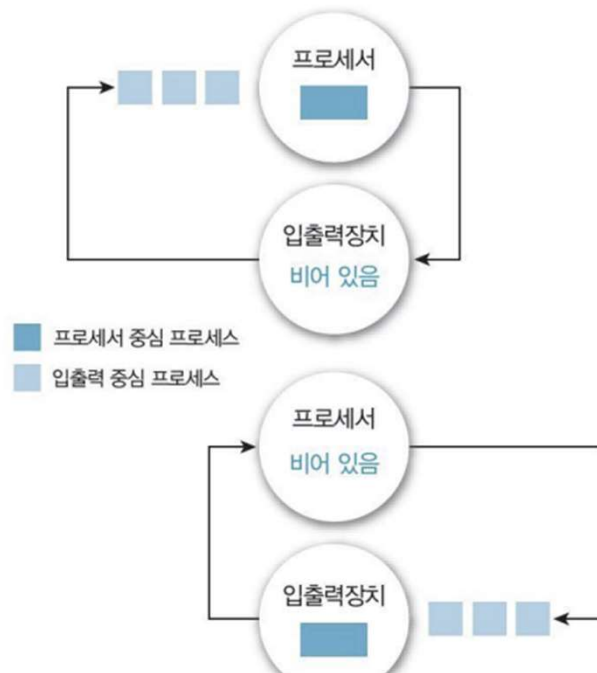
선입선출FIFO 큐로 구현 일괄 처리 시스템에서는 매우 효율적이나 빠른 응답을 요청하는 대 화식 시스템에는 적합하지 않음

- 원리



스케줄링 알고리즘

- 동적 상황에서의 성능



Convoy effect: 프로세서 중심 프로세스 하나가 프로세서를 떠나기를 기다리는 현상, 어느 프로세스 한쪽에 치우쳐 기다리는 불균형 상태.

스케줄링 알고리즘

- 선입선처리 스케줄링의 장점과 단점

장점: 스케줄링의 이해와 구현이 단순하다.

준비 큐에 있는 모든 프로세스가 결국 실행되므로 기아가 없다.

프로세서가 지속적으로 유용한 프로세스를 수행하여 처리율 높다.

단점: 비선점식이므로 대화식 작업에는 부적합.

장기 실행 프로세스가 뒤의 작업을 모두 지연시켜 평균 대기시간이 길어져 최악의 대기시간이 된다.

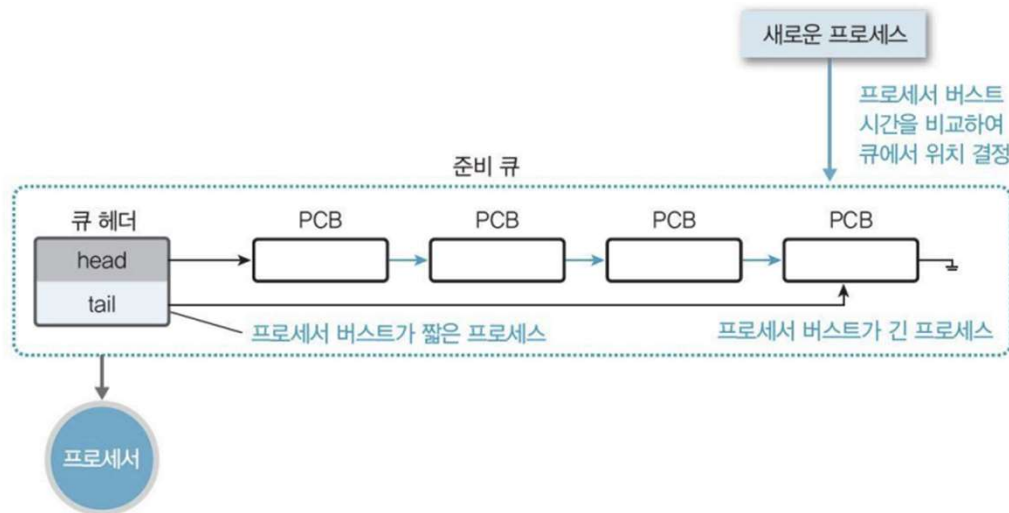
긴 호위 효과가 발생할 수 있다.

스케줄링 알고리즘

- 최소작업 우선 스케줄링(Shortest Job First)

각 작업의 프로세서 실행 시간을 이용하여 프로세서가 사용 가능할 때 실행 시간이 가장 짧은 작업(프로세스)에 할당하는 방법

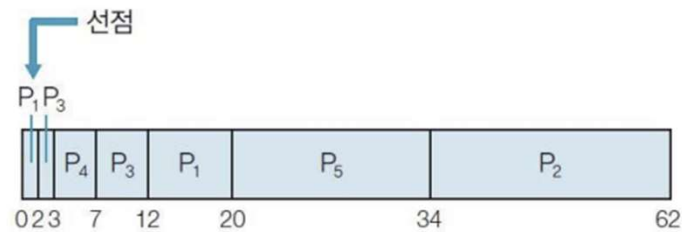
- 원리



스케줄링 알고리즘

• 선점 최소작업 우선 스케줄링

프로세스	도착 시간	실행 시간
P ₁	0	10
P ₂	1	28
P ₃	2	6
P ₄	3	4
P ₅	4	14



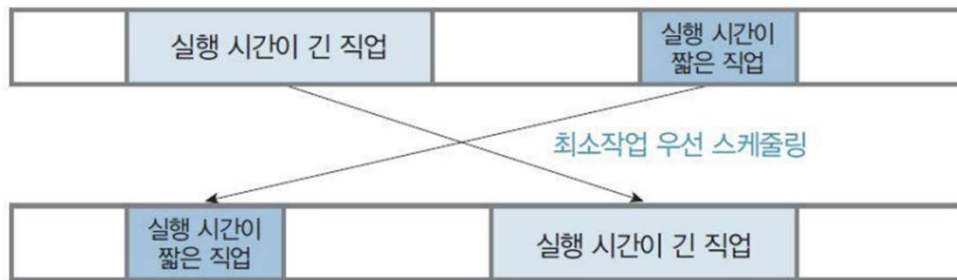
프로세스	반환시간	대기시간
P ₁	$20 - 0 = 20$	$(12 - 2) = 10$
P ₂	$(62 - 1) = 61$	$(34 - 1) = 33$
P ₃	$12 - 2 = 10$	$(7 - 3) = 4$
P ₄	$(7 - 3) = 4$	$(3 - 3) = 0$
P ₅	$(34 - 4) = 30$	$(20 - 4) = 16$

■ 평균 반환시간: $25[(20+61+10+4+30)/5]$

평균 대기시간: $12.6[(10+33+4+0+16)/5]$

스케줄링 알고리즘

- 최소작업 우선 스케줄링의 장점과 단점



장점: 항상 실행 시간이 짧은 작업을 처리하므로 평균 대기시간이 가장 짧다.

단점: 초기의 긴 작업을 짧은 작업을 종료할 때까지 대기시켜 기아가 발생

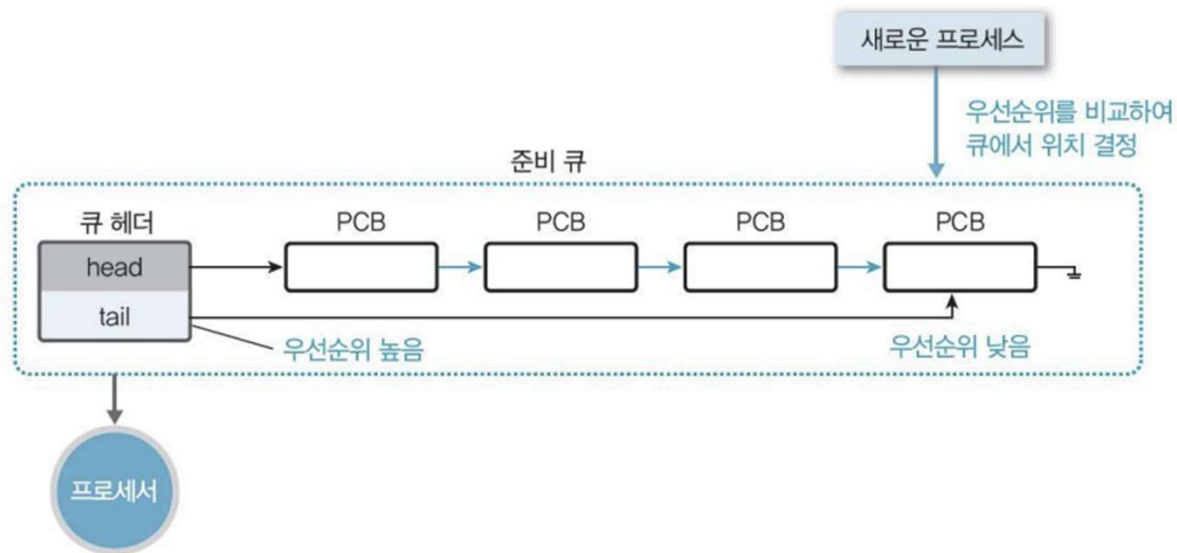
기본적으로 짧은 작업이 항상 실행되도록 설정하므로 불공정한 작업 실행.

실행 시간이 예측하기가 어려워 실용적이지 못하다.

스케줄링 알고리즘

- 우선순위 스케줄링의 개념

우선순위가 동일한 프로세스들은 선입선처리 순서로 스케줄링

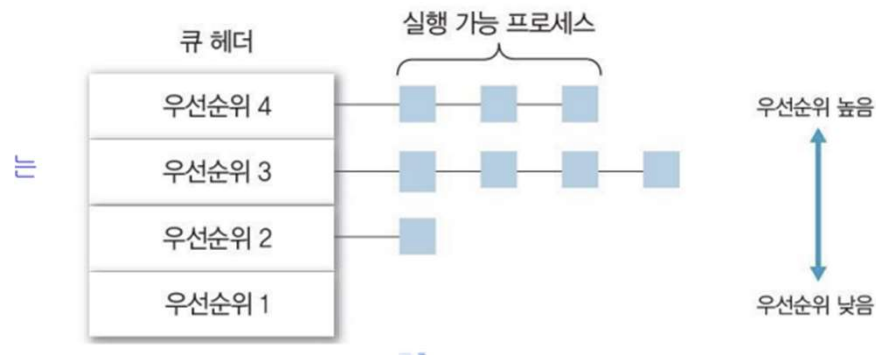


스케줄링 알고리즘

우선순위 스케줄링

내부적 우선순위: 제한 시간, 기억장소 요청량, 사용 파일 수 등

외부적 우선순위: 프로세스 중요성, 사용료 많이 내난 사용자, 작업을 지원하는 부서, 정책적인 요인 등



스케줄링 알고리즘

- 우선 순위 스케줄리의 장점과 단점

장점: 각 프로세스의 상대적 중요성을 정확히 정의할 수 있어 좋다.

다양한 반응으로 실시간 시스템에 사용 가능하다.

단점: 높은 우선순위 프로세스가 프로세서를 많이 사용하면 우선순위가 낮은 프로세스는 무한정 연기되는 기아가 발생할 수 있다.

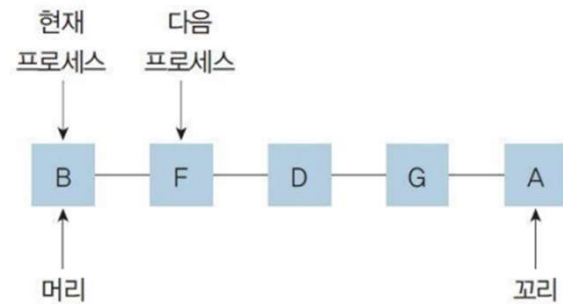
스케줄링 알고리즘

- 라운드 로빈 스케줄링

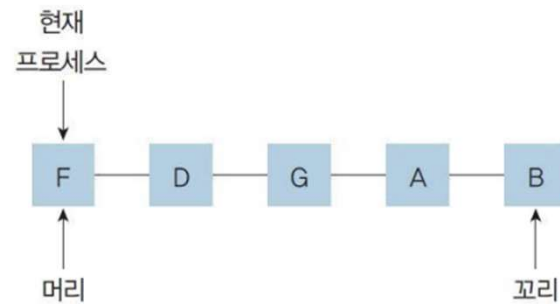
특별히 시분하 시스템을 위해 설계

작은 단위의 시간인 규정 시간량(time quantum) 또는
시간 할당량(time slice) 정의

준비 큐를 순환 큐로 설계하여 스케줄러가 준비 큐를 돌아가면서
한 번에 한 프로세스에 정의된 규정 시간량만큼 프로세서 제공



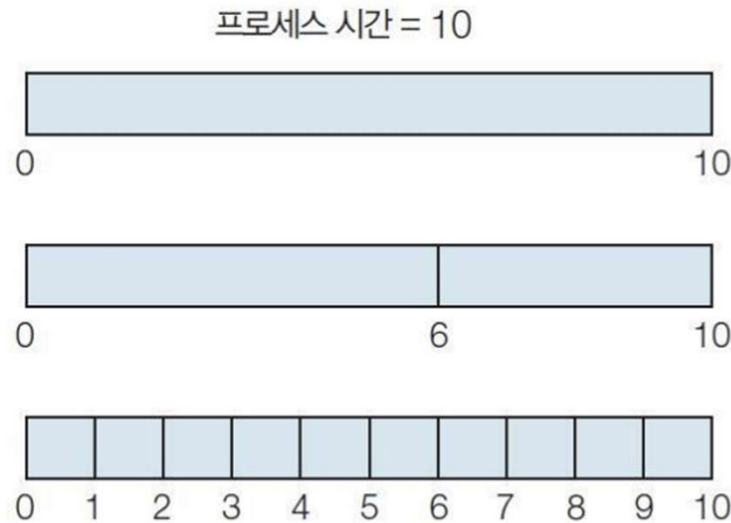
(a) 실행 전



(b) 실행 후

스케줄링 알고리즘

- 문맥 교환 시간이 라운드 로빈 스케줄링에 미치는 영향
Time Slice와 turn around time이 항상
linear 관계는 아니다.



time slice	문맥 교환 횟수
12	0
6	1
1	9

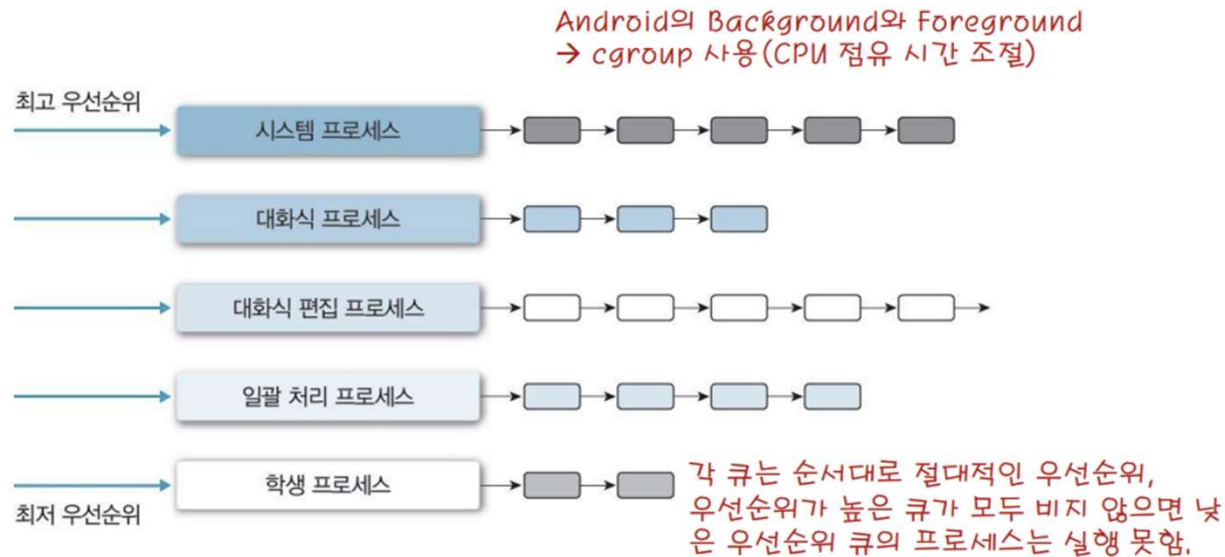
스케줄링 알고리즘

- 다단계 큐 스케줄링(MultiLevel Queue)

각 작업을 서로 다른 묶음으로 분류할 수 있을 때 사용

준비 상태 큐를 종류별로 여러 단계로 분할, 그리고 작업을 메모리의 크기나 프로세스의 형태에 따라 특정 큐에 지정

원리



스케줄링 알고리즘

- 다단계 큐 스케줄링의 장점과 단점

장점: 응답이 빠르다.

단점: 여러 준비 큐와 스케줄링 알고리즘 때문에 추가 오버헤드 발생
우선순위가 낮은 큐의 프로세스는 무한정 대기하는 기아 발생

스케줄링 알고리즘

• 다단계 피드백 큐 스케줄링(MultiLevel Feedback Queue)

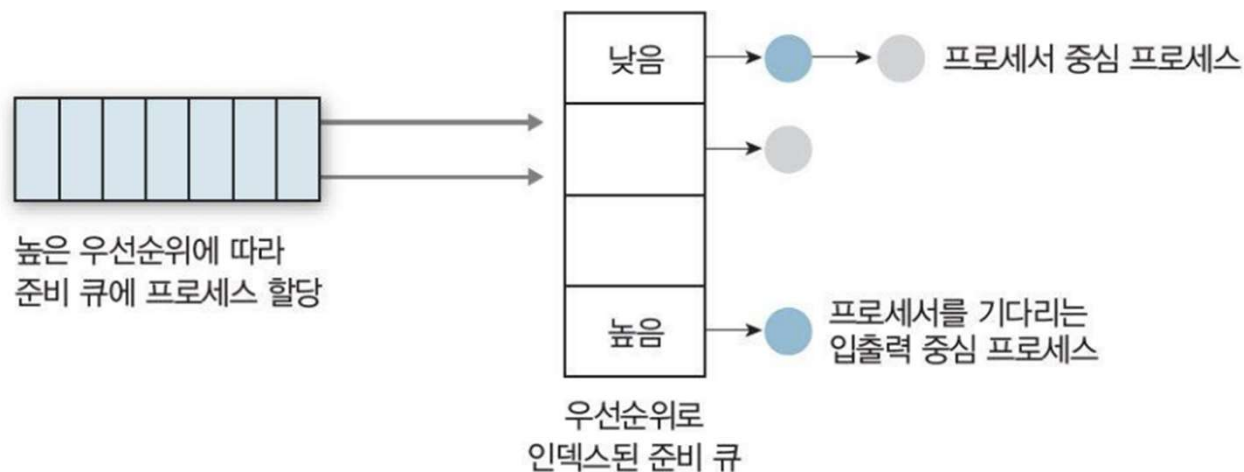
다단계 큐^{MLQ}는 프로세스가 한 큐에서 고정되어 실행

- 스케줄링 부담이 적다는 장점이 있으나 융통성이 떨어진다는 단점

다단계 피드백 큐에서는 프로세스가 큐 사이 이동 가능

- 프로세서 버스트의 특성에 따라 분리

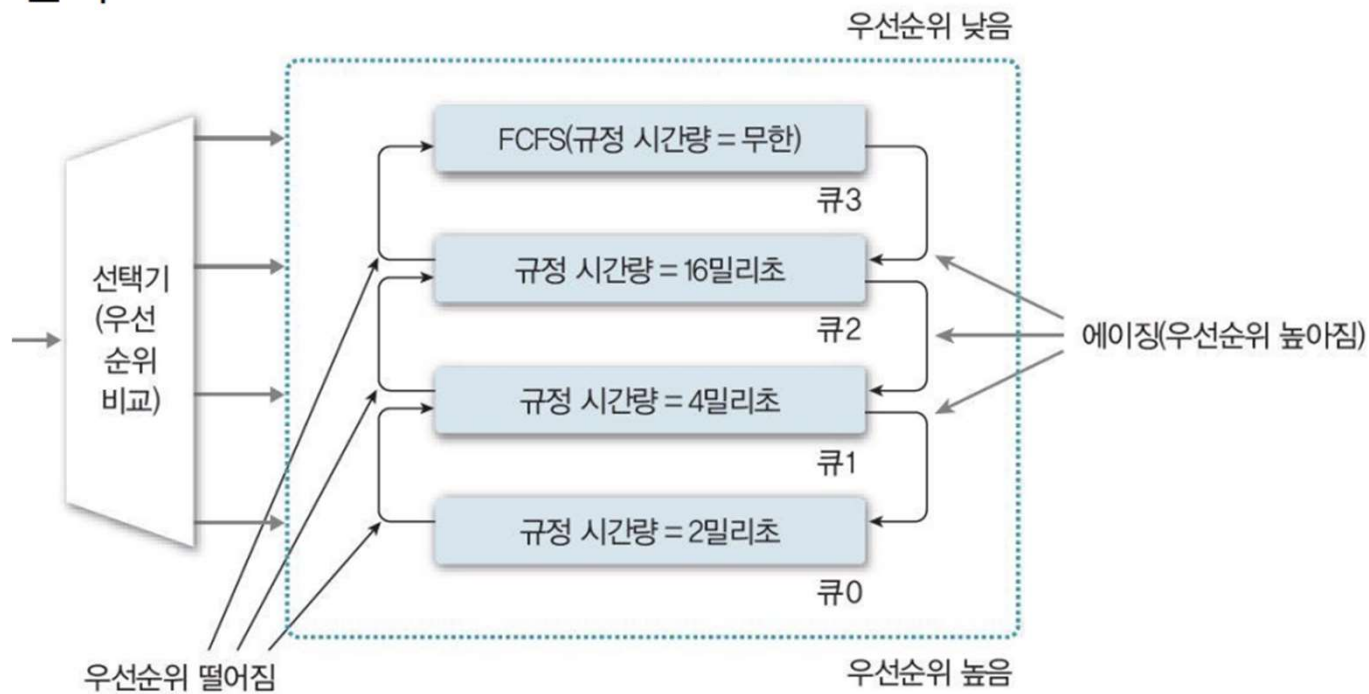
구조



스케줄링 알고리즘

- 다단계 피드백 큐 스케줄링

원리



스케줄링 알고리즘

- 다단계 피드백 큐 스케줄링에서 고려사항

큐_{queue} 수

각 큐에 대한 스케줄링

작업을 좀 더 높은 우선순위의 큐로 격상시키는 시기를 결정하는 방법

작업을 좀 더 낮은 우선순위의 큐로 격하시키는 시기를 결정하는 방법

프로세스들이 어느 큐에 들어갈 것인지 결정하는 방법

프로세스가 서비스를 받는 시기를 결정하는 방법

스케줄링 알고리즘

- 다단계 피드백 큐 스케줄링의 장점과 단점

장점: 매우 유연하여 스케줄러를 특정 시스템에 맞게 구성 가능

자동으로 입출력 중심과 프로세서 중심으로 분류

적응성이 좋아 프로세스의 사전 정보 없어도 최소작업 우선
스케줄링 효과를 보여준다.

단점: 설계와 구현이 매우 복잡하다.

스케줄링 알고리즘

- HRN(Highest Response-ratio Next)

최소작업 우선 스케줄링의 약점인 긴 작업과 짧은 작업 간의 지나친 불평등을 보완

비선점 스케줄링이며 우선순위 스케줄링의 또 다른 예
선입선처리 스케줄링과 최소작업 우선 스케줄링의 약점을 해결
위해 제안.

우선순위가 가변

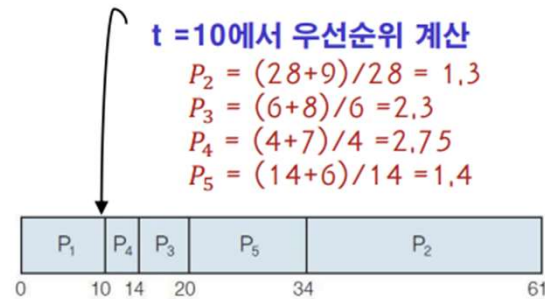
$$\text{우선순위} = \frac{\text{서비스를 받을 시간} + \text{대기한 시간}}{\text{서비스를 받을 시간}}$$

시스템 응답시간 : 대기한 시간 + 서비스를 받을 시간

스케줄링 알고리즘

• HRN 스케줄링 (비선점)

프로세스	도착 시간	실행 시간
P ₁	0	10
P ₂	1	28
P ₃	2	6
P ₄	3	4
P ₅	4	14



프로세스	반환시간	대기시간
P ₁	10	0
P ₂	(62 - 1) = 61	(34 - 1) = 33
P ₃	(20 - 2) = 18	(14 - 2) = 12
P ₄	(14 - 3) = 11	(10 - 3) = 7
P ₅	(34 - 4) = 30	(20 - 4) = 16
평균 반환시간: 26 [= (10 + 61 + 18 + 11 + 30)/5]		평균 대기시간: 14 [= (0 + 33 + 12 + 7 + 16)/5]

스케줄링 알고리즘

- HRN 스케줄링의 장점과 단점

장점: 자원을 효율적으로 활용한다.

기아가 발생하지 않는다.

단점: 오버헤드가 높을 수 있다.(메모리와 프로세서 낭비)

→ 매 스케줄링 시점마다 우선순위 계산.

Q & A