

운영체제 3주차 2차시

1

선입 선출, (비선점/선점)
최소 작업 우선 스케줄링

선입 선출,(비선점/선점)최소 작업 우선 스케줄링

1 선입선출(FCFS, First Come First Served) 스케줄링의 개념

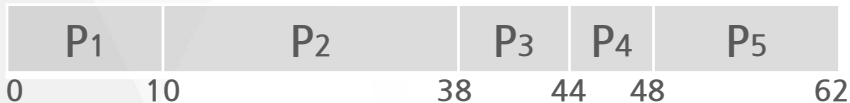
- ▶ 비선점 방법으로 프로세서 스케줄링 알고리즘 중 가장 단순
- ▶ 프로세서 요청하는 순서대로 프로세서 할당, 선입선출 FIFO 큐로 구현
- ▶ 일괄 처리 시스템에서는 매우 효율적이거나 빠른 응답을 요청하는 대화식 시스템에는 적합하지 않음

선입 선출, (비선점/선점) 최소 작업 우선 스케줄링

1 선입선출(FCFS, First Come First Served) 스케줄링의 개념

프로세스	도착시간	실행시간
P1	0	10
P2	1	28
P3	2	6
P4	3	4
P5	4	14

(a)준비 큐



(b)간트 차트

프로세스	반환시간	대기시간
P1	10	0
P2	$(38-1)=37$	$(10-1)=9$
P3	$(44-2)=42$	$(38-2)=36$
P4	$(48-3)=45$	$(44-3)=41$
P5	$(62-4)=58$	$(48-4)=44$

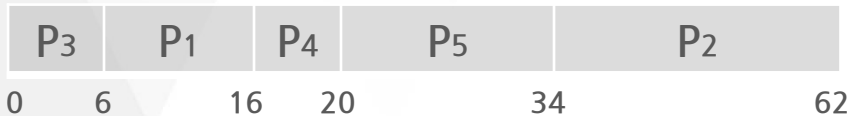
평균반환시간	평균대기시간
$(10+37+42+45+58)/5$	$(0+9+36+41+44)/5$

선입 선출, (비선점/선점) 최소 작업 우선 스케줄링

1 선입 선출(FCFS, First Come First Served) 스케줄링의 개념

프로세스	도착시간	실행시간
P ₃	0	6
P ₁	1	10
P ₄	2	4
P ₅	3	14
P ₂	4	28

(a)준비 큐



(b)간트 차트

프로세스	반환시간	대기시간
P ₁	$(16-1)=15$	$(6-1)=5$
P ₂	$(62-4)=58$	$(34-4)=30$
P ₃	6	0
P ₄	$(20-2)=18$	$(16-2)=14$
P ₅	$(34-3)=31$	$(20-3)=17$

평균반환시간	평균대기시간
25.6	13.2

선입 선출, (비선점/선점) 최소 작업 우선 스케줄링

1 선입선출(FCFS, First Come First Served) 스케줄링의 개념

장점

- 스케줄링의 이해와 구현이 단순함
- 준비 큐에 있는 모든 프로세스가 결국 실행되므로 기아 없는 공정한 정책임
- 프로세서가 지속적으로 유용한 프로세스를 수행하여 처리율이 높음

단점

- 비선점식이므로 대화식 프로세스에는 부적합함

1

선입 선출, (비선점/선점) 최소 작업 우선 스케줄링

2 최소작업 우선(SJF, Shortest Job First) 스케줄링 개념

- ▶ 준비큐에 있는 프로세스 중에서 실행 시간이 가장 짧은 작업부터 CPU를 할당하는 비선점형 방식
- ▶ 최단 작업 우선 스케줄링이라고도 함

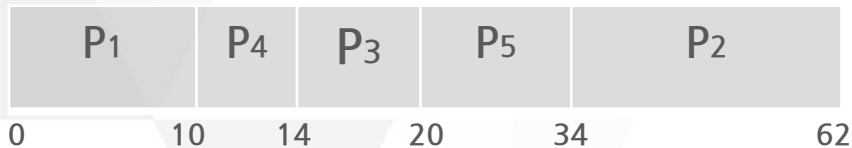
1

선입 선출,(비선점/선점) 최소 작업 우선 스케줄링

2 최소작업 우선(SJF, Shortest Job First) 스케줄링 개념

프로세스	도착시간	실행시간
P1	0	10
P2	1	28
P3	2	6
P4	3	4
P5	4	14

(a)준비 큐



(b)간트 차트

프로세스	반환시간	대기시간
P1	10	0
P2	$(62-1)=61$	$(34-1)=33$
P3	$(20-2)=18$	$(14-2)=12$
P4	$(14-3)=11$	$(10-3)=7$
P5	$(34-4)=30$	$(20-4)=16$

평균반환시간	평균대기시간
26	13.6

1

선입 선출,(비선점/선점) 최소 작업 우선 스케줄링

2 최소작업 우선(SJF, Shortest Job First) 스케줄링 개념

장점

- 항상 실행 시간이 짧은 작업을 신속하게 실행하므로 평균 대기 시간이 가장 짧음

단점

- 초기의 긴 작업을 짧은 작업을 종료할 때까지 대기시켜 기아가 발생
- 기본적으로 짧은 작업이 항상 실행되도록 설정하므로 불공정한 작업을 실행

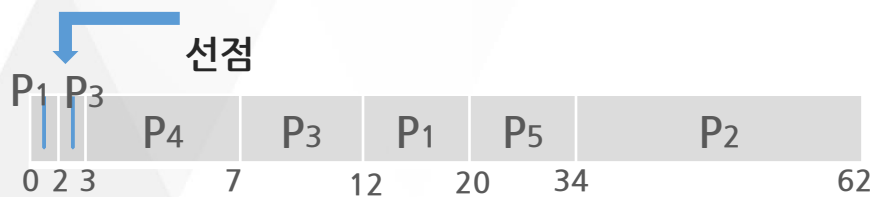
선입 선출,(비선점/선점) 최소 작업 우선 스케줄링

3 선점 최소작업 우선(SRTF, Shortest Remaining Time First)

프로세스	도착시간	실행시간
P1	0	10
P2	1	28
P3	2	6
P4	3	4
P5	4	14

프로세스	반환시간	대기시간
P1	$(20-2)=18$	$(12-2)=10$
P2	$(62-1)=61$	$(34-1)=33$
P3	$(12-3)=9$	$(7-3)=4$
P4	$(7-3)=4$	$(3-3)=0$
P5	$(34-4)=30$	$(20-4)=16$

(a)준비 큐



(b)간트 차트

평균반환시간	평균대기시간
24.4	12.6

2 우선순위, 순환(라운드로빈) 할당 스케줄링

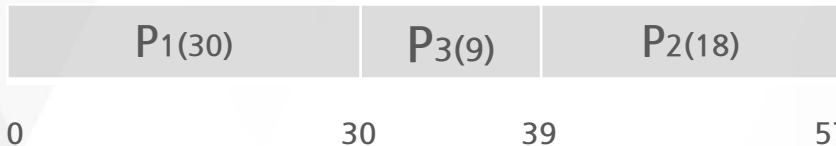
우선순위, 순환 할당(라운드로빈) 스케줄링

1 우선순위 스케줄링(Priority Scheduling)의 개념

▶ 프로세스의 중요도에 따른 우선순위를 반영한 스케줄링 알고리즘

▶

프로세스	도착시간	실행시간	우선순위
P ₁	0	30	3
P ₂	3	18	2
P ₃	6	9	1



▶ 우선순위 스케줄링도 선점 또는 비선점 가능

우선순위, 순환 할당(라운드로빈) 스케줄링

1 우선순위 스케줄링(Priority Scheduling)의 개념

◆ 고정 우선순위 알고리즘

- 한 번 우선순위를 부여 받으면 종료될 때까지 우선순위가 고정
- 단순하게 구현할 수 있지만 시시각각 변하는 시스템의 상황을 반영하지 못해 효율성이 떨어짐

◆ 변동 우선순위 알고리즘

- 일정 시간마다 우선순위가 변하여 일정 시간마다 우선 순위를 새로 계산하고 이를 반영
- 복잡하지만 시스템의 상황을 반영하여 효율적인 운영 가능

2 우선순위 스케줄링(Priority Scheduling)의 문제점

준비 큐에 있는 프로세스의 순서를 무시하고 우선순위가 높은 프로세스에 먼저 CPU를 할당하므로 공평성을 위배하고 기아 현상을 일으킴

◆ 해결방법 ⇨ 에이징

■ 에이징

- ❖ 시스템에서 오래 대기하는 프로세스들의 우선순위를 점진적으로 증가시키는 방법
- ❖ 시간이 지나면 점차 프로세스의 우선순위 높아짐

3 우선순위 스케줄링(Priority Scheduling)의 장점과 단점

장점

- 각 프로세서의 상대적 중요성을 정확히 정의할 수 있음

단점

- 높은 우선순위 프로세스가 프로세서를 많이 사용하면 우선순위가 낮은 프로세스는 무한정 연기되는 기아가 발생할 수 있음

- ▶ 한 프로세스가 할당받은 시간(타임슬라이스) 동안 작업을 하다가 작업을 완료하지 못하면 준비 큐의 맨 뒤로 가서 자기 차례를 기다리는 방식
- ▶ 프로세스들이 작업을 완료할 때까지 계속 순환하면서 실행

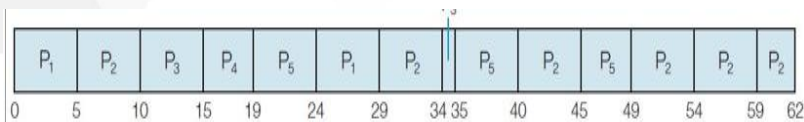
4 라운드 로빈 스케줄링 개념

프로세스	도착시간	실행시간
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	29	(24-5)
P ₂	(62-1)	(49-15-1)
P ₃	(35-2)	(34-5-2)
P ₄	(19-3)	(15-3)
P ₅	(49-4)	(45-10-4)

타임 슬라이스를 5 가정해보자

간트 도표 그려보기



평균반환시간

36.8

평균대기시간

23.4

4 라운드 로빈 스케줄링 개념

장점

- 시분할 방식의 시스템에서 효과적임

단점

- 할당 시간이 크면 FCFS 방식과 같음
- 할당 시간이 작으면 자주 문맥교환이 발생하므로 오버헤드가 커짐

3 다단계 큐, 다단계 피드백 큐, HRN 스케줄링

다단계 큐, 다단계 피드백 큐, HRN 스케줄링

1 다단계 큐 스케줄링의 개념

- ▶ 우선 순위에 따라 준비 큐를 여러 개 사용하는 방식
- ▶ 프로세스는 운영체제로부터 부여받은 우선 순위에 따라 해당 우선순위의 큐에 삽입
- ▶ 준비 상태 큐를 종류별로 여러 단계로 분할, 그리고 작업을 메모리의 크기나 프로세스의 형태에 따라 특정 큐에 지정
 - 각 큐는 자신만의 독자적인 스케줄링 갖음

장점

- 응답이 빠름

단점

- 여러 준비 큐와 스케줄링 알고리즘 때문에 추가 오버헤드가 발생
- 우선순위가 낮은 큐의 프로세스는 무한정 대기하는 기아가 발생할 수 있음

3 다단계 피드백 큐 스케줄링의 개념

- ▶ 프로세스가 CPU를 한 번씩 할당받아 실행될 때마다 프로세스의 우선순위를 낮춤으로써, 다단계 큐에서 우선 순위가 낮은 프로세스의 실행이 연기되는 문제를 완화
- ▶ 우선순위에 따라 타임 슬라이스의 크기가 다름
- ▶ 우선순위가 낮아질수록 CPU를 얻을 확률이 적어짐. 따라서 한번 CPU를 잡을 때 많이 작업하기 위해 낮은 우선순위의 타임 슬라이스를 크게 지정함

장점

- 매우 유연하여 스케줄러를 특정 시스템에 맞게 구성할 수 있음
- 자동으로 입출력 중심과 프로세서 중심 프로세스 분류
- 적응성이 좋아 프로세서의 사전 정보 정보가 없어도 최소작업 우선 스케줄링의 효과를 보여줌

단점

- 설계와 구현이 매우 복잡함

5 HRN 스케줄링의 개념

- ▶ 최소작업 우선 스케줄링의 약점인 긴 작업과 짧은 작업 간의 지나친 불평 등을 보완
- ▶ 비선점 스케줄링이며 우선순위 스케줄링의 또 다른 예
- ▶ 선입선처리 스케줄링과 최소작업 우선 스케줄링의 약점을 해결 위해 제안
- ▶ 서비스를 받기 위해 기다린 시간과 CPU 사용 시간을 고려하여 스케줄링을 하는 방식

▶ 우선순위

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

▶ 시스템 응답시간

시스템 응답시간 : 대기한 시간 + 서비스를 받을 시간

프로세스	도착시간	실행시간
P1	0	10
P2	1	28
P3	2	6
P4	3	4
P5	4	14

(a)준비 큐



(b)간트 차트

프로세스	반환시간	대기시간
P1	10	0
P2	$62-1=61$	$34-1=33$
P3	$20-2=18$	$14-2=12$
P4	$14-3=11$	$10-3=7$
P5	$34-4=30$	$20-4=16$

평균반환시간	평균대기시간
26	14

6 HRN 스케줄링의 장점과 단점

장점

- 실행 시간이 짧은 프로세스의 우선순위를 높게 설정하면서도 대기 시간을 고려하여 기아 현상을 완화

단점

- 오버헤드가 높을 수 있음 (메모리와 프로세서 낭비)