

CP5. CPU 스케줄링

스케줄링 < ① 자원내 스케줄링
② 컴퓨터내 스케줄링

다중프로그래밍
(도입) CPU 유휴시간 ↓ ⇒ 활용을 ↑
→ 하면서 여러 스케줄링 도입.
① 작업 스케줄링
② CPU "

CPU burst 프로그램 실행중 CPU에서 중심인 실행
I/O " " I/O 대기 중심인 입출력 "

프로그램 실행 순서 CPU burst - I/O burst - 빈번.

CPU 스케줄링 실행 준비상태 READY의 프로세스 중 하나를 선택해서 실행.
→ (예) CPU 활용률 ⇒ 자원 활용량.

CPU 스케줄링 기준	① CPU 활용률	⑤ 대기시간	P, 11, 12
	② 처리량	⑥ 소요시간	
(11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50) (51) (52) (53) (54) (55) (56) (57) (58) (59) (60) (61) (62) (63) (64) (65) (66) (67) (68) (69) (70) (71) (72) (73) (74) (75) (76) (77) (78) (79) (80) (81) (82) (83) (84) (85) (86) (87) (88) (89) (90) (91) (92) (93) (94) (95) (96) (97) (98) (99) (100)	③ 공정성	⑦ 시스템장착	
목표가치	④ 응답시간	⑧ 자원 활용률	

대부분 문제 - 너무 빨리 리미트시간 CPU 사용 X.
타임슬라이스 : 스케줄링 스케줄러가 한 번 할당하는 CPU 시간
커널이 스케줄링 단행하는 주기시간
컨텍스트 스위칭과 관련시간이 길어야 함.
→ 타임슬라이스마다 스케줄(프로그램)이
CPU가 변경 - 이때마다 컨텍스트 스위칭 발생

2. CPU 스케줄링 기법

" 실행
시스템콜

1. 12도가 시스템콜 끝이 I/O를 요청하여 블록
2. // 자발적으로 CPU 반환
3. // 이 타임슬라이스 소진 → 타이머 인터럽트 발생
4. 더 높은 우선의 12도가 요청한 압출력 작업으로 → 인터럽트 발생

스케줄링을 담당하는 커널 스케줄러(또는 프로세스)는 많은

커널? 커널 코드의 일부.

스케줄링 코드 실행 - 시스템콜을 @ ISR이 끝나는 미시마이크로

디스패치: 커널 스케줄러로 실행하는 커널

스케줄링 & 디스패치 모두 실행 시간이 짧도록 구성

(숙제)

스케줄링 실행 → 인터럽트 @ 시스템콜 처리 → CPU 스케줄링 → 디스패치 → 스레드 B

<강제 중단>

비선점

선점

현재 실행 중인 스레드를 강제 중단 X (Sleep or I/O 요청을 위해)
강제 중단 O - 폴링 사용.

기아

에이징

스케줄링 스케줄링에 선택받지 못한 채 오래 준비 상태에 있음..

기아에 대한, 스레드가 준비 리스트에 있는 동안,
머리는 시간이 지남에 따라 스케줄링 순위를 높임.

3. OPUS 스케줄링 알고리즘

비선점 FCFS, SJF.

선점 SRTF, RR, MLFQ.

동태가능 4가지...

FCFS

선입선처리. 먼저 도착한 프로세스 먼저 스케줄링

파라미터: 스케줄링 도착시간

우선순위: X.

기아: X (프로세스가 들어왔을 때, 뒤로 밀려나기만 할 뿐)

성능 이슈: 처리율 ↓, 회전 지연 발생

SJF

Shortest

Job first

최단작업 우선 스케줄링. 예상시간이 짧은 것 선택

파라미터: 예상시간. (근데 현실적이지 않음)

우선순위: X.

기아: 발생 가능. (짧은 게 먼저 들어오면, 긴 스케줄링이?)

성능: 평균 대기시간 최소화.

문제: 실행시간 예측 불가능으로 처리량 감소

Priority

선점/비선점

우선순위에 따라 스케줄링 할당시키기 위한 목적

가장 높은 순위의 프로세스 선택

파라미터: 스케줄링 고정 우선 순위.

우선순위: 0

기아: ○. 낮은 순위의 프로세스가 들어오면, 스케줄링이 어려움.

성능: 높은 우선 순위 프로세스를 대기 - 응답 시간 감소

특징: 스케줄링 고정 우선 순위를 가지는 실시간 시스템에서 사용

SJF의 단점.

SRTF

Shortest

Remaining
Time

First.

최소 잔여 시간 우선 스케줄링. 모든 실행 시간이 가장 짧은.

특이점: 다중 실행 시간과 높은 실행 시간

우선순위: X.

기아: 발생 가능

성능: $\frac{\text{평균 대기 시간}}{\text{SJF}}$ 같음.

문제점: SJF 같음

RR

Round

Robin

균형 실행 계획으로 주기적이며, 큐에 대기 중인 스레드를

타임슬라이스 주기로 돌려서 실행. 회전하는 큐를 이용.

호출 순서대로 큐에 삽입.

특이점: 타임슬라이스.

우선순위: X.

기아: X. 우선순위 X, 타임슬라이스 정해져있어 한 번 실행 후 반드시 실행

성능: 균형. 기아 X. 극단 없음.

같은 스케줄링으로 오버헤드. (타임슬라이스 줄 때 더 좋음)

케이스도 속칭으로 가능함.

균형의 차이는 타임슬라이스 \rightarrow FCFS

// \downarrow = SJF / SRTF

MLQ

Multi

level

Queue

선/배치

- 스레드를 n 개의 우선순위 레벨로 구분,

레벨이 높을수록 스레드 우선 처리.

- 고정된 n 개의 큐 사용, 각각에 스케줄링.

특이점: 스레드별 고정 우선순위.

우선순위: 0

기아: 0 높은 우선 //

성능: 스레드의 고정 순위를 가진 시스템에서 사용.

시스템 스레드, 대화식 스레드, 등로 나뉘고 사용

~~기아나 타임슬라이스를 가짐. 우선순위가 높을수록 타임슬라이스도 적음.~~

p 173

MLFQ - 기아를 없애기 위해, 여러레벨이 섞이어서
Multi 스레드 이용 가능하도록 설계
Level - 주요 타임슬라이스가 있고,
Feedback 높은 레벨이면 타임슬라이스가 짧아짐.
QV - 회기이다; 각 큐의 타임슬라이스.
선택 우선순위: X (에너지가 없음)
기아: X → 대기시간 오래되면, 더 높은 레벨이 되어 이득
성능: 선택 우선순위 스텝드 큐에서 실행 → (다바를 줄 수 있음)

4. 실시간 시스템에서의 CPU 스케줄링

※ 실용적 CPU 사용시 문제점.

① 컨텍스트 스위칭 오버헤드 증가 문제

⇒ CPU 리타이밍 비용, 고정된 스레드 유지 비용

② 코어별 부하 불균형 문제

⇒ 후시 마이그레이션 / 폴 마이그레이션