

20220550 윤명섭

컴퓨터 운영체제 5주차 과제

연습문제 27

4번

연습문제 28

리눅스 CFS(Completely Fair Scheduler)가 RR(Round Robin)과 달리, 스케줄링 시마다 타임 슬라이스를 재계산하는 이유는 공정성을 극대화하고 시스템의 효율성을 높이기 위해서입니다.

구체적인 이유는 다음과 같습니다.

### 공정성 확보

**가변적인 작업 부하 처리:** 시스템의 작업 부하는 시간에 따라 크게 변할 수 있습니다.

고정된 타임 슬라이스를 사용하는 RR 방식은 작업 부하가 적을 때는 불필요한 문맥 교환을 일으켜 성능 저하를 일으키고

, 작업 부하가 클 때는 특정 프로세스가 CPU를 독점하는 현상을 초래할 수 있습니다.

**가상 실행 시간(Virtual Run time) 개념:** CFS는 각 프로세스의 가상 실행 시간을 추적하고, 이 값을 기준으로 다음 실행 프로세스를 결정합니다. 가상 실행 시간은 프로세스의 우선순위를 고려하여 계산되며, 우선순위가 높은 프로세스는 가상 실행 시간이 느리게 증가합니다.

**공정한 CPU 할당:** CFS는 모든 프로세스에게 공평하게 CPU 시간을 할당하려고 노력합니다. 준비 큐에 있는 프로세스 수에 따라 타임 슬라이스를 동적으로 조정함으로써, 각 프로세스가 자신의 우선순위에 비례하여 CPU 시간을 할당받도록 보장합니다.

### 시스템 효율성 향상

**문맥 교환 오버헤드 최소화:** 타임 슬라이스를 너무 작게 설정하면, 문맥 교환이 빈번하게

발생하여 시스템 성능을 저하시킬 수 있습니다. CFS는 작업 부하에 따라 적절한 타임 슬라이스를 계산하여 문맥 교환 오버헤드를 최소화합니다.

**대화형 작업 성능 향상:** 대화형 작업은 짧은 시간 동안 CPU를 사용하고 대기하는 패턴을 보입니다. CFS는 이러한 작업의 특성을 고려하여 짧은 타임 슬라이스를 할당하고, 빠른 응답성을 제공합니다.

**배치 작업 처리 효율성 향상:** 배치 작업은 오랜 시간 동안 CPU를 사용하는 패턴을 보입니다. CFS는 이러한 작업에 대해서는 상대적으로 긴 타임 슬라이스를 할당하여 전체 시스템의 처리량을 높입니다.

요약하면, 리눅스 CFS는 고정된 타임 슬라이스를 사용하는 RR 방식의 단점을 극복하고, 다양한 작업 부하 상황에서 공정성과 효율성을 모두 확보하기 위해 동적으로 타임 슬라이스를 계산합니다. 이를 통해 사용자에게 더 나은 시스템 응답성과 성능을 제공합니다.