운영체제 Project 1

202111057 문서흔

1. 변수 추가

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

Proc.h에 먼저 Little core를 0으로, Big core를 1로 정의.

폰트, 텍스트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

Struct proc에 프로세스가 할당된 CPU를 나타내는 cpu\_affinity, 프로세스 누적 실행 시간을 기록하는 runtime 변수를 추가해줬다.

1. RR cycle: 10ms로 가정  
   
2. Big Core 이동

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* Little Core의 RUNNABLE 상태 프로세스 수를 계산:
  + runnable\_count 변수를 통해 Little Core에 할당된 RUNNABLE 상태의 프로세스 수를 count
  + 만약 RUNNABLE 상태이고, cpu\_affinity가 LITTLE\_CORE인 프로세스가 있다면 runnable\_count를 증가시킵니다.
* 가장 오래 실행된 프로세스를 찾음:
  + heavy\_proc 변수를 사용하여 가장 오래 실행된 프로세스를 선택.
  + p->runtime을 기준으로 실행 시간이 가장 긴 프로세스를 heavy\_proc에 저장
* Little Core의 부하가 임계치를 초과하면 프로세스를 이동:
  + runnable\_count가 5보다 크면, 즉 Little Core의 부하가 높을 경우, migrate\_process(heavy\_proc)를 호출하여 가장 무거운 프로세스(가장 오래 실행된 프로세스)를 Big Core로 이동
  + Big Core 이동 함수: migrate\_process  
    선택된 프로세스를 Big Core로 이동  
    텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명

1. 결과

* 프로세스 수: 총 24개의 프로세스가 생성.
* Response time
  + 대부분 1. 5이상인 경우, 이전에 이미 많은 프로세스가 실행 중이어서 그런 것으로 분석됨.
  + Average response time: 2
* Turnaround time
  + Average turnaround time: 555
* Core 이동: 대부분 Little에서 Big으로 이동.  
  텍스트, 스크린샷, 흑백, 모노크롬이(가) 표시된 사진

  자동 생성된 설명