스케줄러 설계

OS

학번 : 12151487

이름 : 조규영(GyuYoung Cho)

**1. 개요**

Pthread를 사용해 Multi thread 기반으로 scheduling을 해봄.

**2. 구현한 스케줄러 설명**

- round-robin scheduling

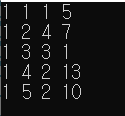
Quantum을 주고 quantum할당량만큼 process들이 작업을 돌아가면서 함. Round robin 스케줄러가 우선순위나 다른 스케줄러보다 process작업 분배가 효율적이라고 생각해서 구현해 봄.

**3. 구현 방법**

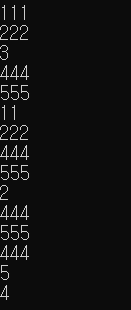
|  |  |
| --- | --- |
|  | **- 환경 변수 설정 및 라이브러리 로드**  **- thread 개수는 5개** |
|  | **- input을 받는 data type 구조체 설정**  **- 시간을 전역으로 설정한 times, thread를 결정하는 turn, process 개수, data타입 배열 설정** |
|  | **- 다음 thread를 결정하는 함수. for문의 시작 번호는 현재 thread의 다음 번호이며 할당량이 있으면 turn에 다음 번호가 지정됨. 할당량이 0이면 반복문이 계속 돌아가며 그 다음 번호가 계속 할당되고 자기 번호로 돌아오면 남은 할당량이 모두 0이라 모든 thread가 끝났음을 의미함.** |
|  | **- pthread\_create에 들어가는 함수. quantum을 주고 process\_id를 받음.**  **- while문에 들어가 process가 끝날 때까지 반복(time[i]==0)**  **- 두 번째 while문에서 자기 차례가 아니면 무한 루프로 들어감.**  **- if문에서 quantum보다 할당량이 많다면 할당량에서 quantum을 뺌** |
|  | **- 남은 양이 quantum보다 작다면 남은 할당량이 0이됨. 다음 thread를 정하고 나서 전체 while을 반복문을 벗어날 수 있게 되고 thread가 종료됨.** |
|  | **- Main 함수. Thread 개수를 설정하고 Input을 입력받는다.** |
|  | **- 5개의 thread 생성. 반복문을 이용하여 5개의 thread를 기다리는 전략 사용.** |

**4. 출력 결과**

input



Quantum=3일 경우



- quantum이 3이기 때문에 pid가 1인 프로세스는 3만큼 일하게 되고 그 다음 process에 할당됨. 이 때 pid가 1인 프로세스의 남은 할당량은 2이고 작업 순서에서 맨 뒤로 배정됨. Pid가 2인 프로세스가 3만큼 실행되고 그 다음으로 넘어가서 pid가 3인 프로세스는 1만큼 수행되고 종료됨. Pid가 4와 5인 process도 3만큼 수행되고 pid가 1인 process로 다시 넘어간다. Pid가 1,2인 process는 남은 할당량이 2,4이므로 1은 종료되고 2는 계속 남아있다. 4와 5를 수행한 뒤 2가 수행되어 종료되고 4와 5도 남은 할당량을 계속 수행한 뒤 종료하게 된다.

**5. 평가**

Quantum이 3인 경우만 실행해 보았는데 그 이상일 때가 context switch가 적고 효율적일 것으로 예상된다. 너무 크면 round robin을 실행하는 의미가 없기 때문에 적절한 quantum을 찾는 것이 중요하다고 봄.

**기타**

개발 환경

Visual Studio 2019에서 수행. pthread library를 사용하기 위해 환경 설정 작업을 함.