운영체제

과제 #1 ku\_cfs



컴퓨터공학과

201411292

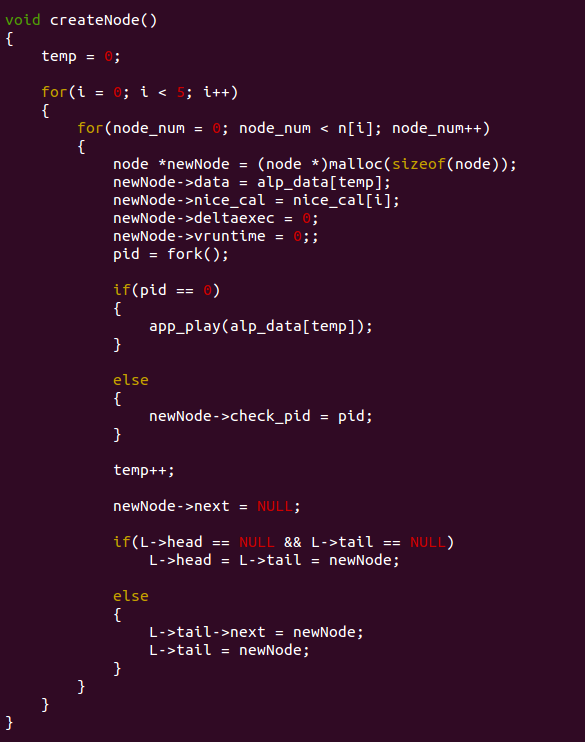
이민환

Submisson

우리가 만들 프로그램은 completely fair처럼 vruntime이 가장 짧은 프로세스에 시그널을 보내 실행시켜주고 time slice 시간이 지난 후 다시 vruntime이 짧은 프로세스를 찾아 실행 시켜주는 것이다. 매개변수로는 n1, n2, n3, n4, n5, ts를 받고 각각의 n들은 차례대로 nice 값 -2, -1, 0, 1, 2를 가지는 프로세스들이며, ts 값 만큼 time slice 개수를 지정해준다.

Basic design

우선, 프로그램을 실행할 때 받은 n1, n2, n3, n4, n5, ts 값을 저장해준다. 그 후, n들의 총합 만큼 노드를 생성하여 필요한 데이터들을 저장해 주고, 자식 프로세스도 만들어 주고, linkedlist 형식으로 연결을 해준다. 이 때, 자식 프로세스를 만들기 위해 fork()를 하면 프로세스들이 여러 개 같이 돌기 때문에, ku\_app을 자식 프로세스에서만 실행을 시켜 주어 멈추게 해준다. 하지만, 코드가 진행이 되면서 ku\_app을 실행시키 전에 다음 단계로 넘어가는 경우가 발생하여, 문제가 일어날 수 있기 때문에 sleep 함수를 사용하여 다음 단계로 가기 전에 잠시 멈추어 준다.



이제 해주어야 할 것은 자식 프로세스들의 vruntime을 계산하여 적은 값의 프로세스를 다시 실행시켜 알파벳이 찍히도록 해야 한다. 프로세스들의 vruntime을 노드 안에 넣어주고, 가장 적은 값이 든 노드를 linkedlist의 제일 앞으로 옮겨준다.



처음 linkedlist를 만들 때는 vruntime이 모두 0이므로 다시 정렬해줄 필요 없이 가장 앞에 있는 노드에 있는 자식프로세스 pid값을 가져와 SIGCONT 신호를 보내주어 프로세스를 실행시킨다. 그리고 1초 뒤 SIGSTOP 신호를 보내 멈추어 준다. 그 후, vruntime을 새로 계산해주어 노드에 넣어주고, vruntime이 가장 적은 값을 가진 노드를 제일 앞으로 옮겨준 후, ts만큼 반복해주면 된다. vruntime을 새로 계산해주기 위해서는 자식 프로세스의 deltaexec을 알아야 한다. 그래서 SIGCONT 신호를 줄때부터 SIGSTOP 신호를 줄때까지의 시간을 deltaexec라고 잡고, clock() 함수를 통해 시간을 변수에 저장하여 노드에 보내준다.

Description for important functions

|  |  |  |
| --- | --- | --- |
| timer\_handler() | Functionality | Time slice를 1초로 설정 한 후 메인에서 pause()를 계속해서 풀어준다. |
| Parameters | None |
| Return Value | None |

|  |  |  |
| --- | --- | --- |
| cont\_handler() | Functionality | vruntime의 값이 가장 작은 값을 가지는 자식프로세스에 SIGCONT 시그널을 보내준다. |
| Parameters | None |
| Return Value | None |

|  |  |  |
| --- | --- | --- |
| stop\_handler() | Functionality | 현재 진행되고 있는 자식프로세스에 SIGSTOP 시그널을 보내준다. |
| Parameters | None |
| Return Value | None |

|  |  |  |
| --- | --- | --- |
| vruntime\_cal() | Functionality | deltaexec값을 새로 받아와 저장해주고 vruntime을 다시 계산해준다. |
| Parameters | float clock\_end |
| Return Value | None |

|  |  |  |
| --- | --- | --- |
| app\_play() | Functionality | Execvp를 통해 ku\_app을 실행시켜준다. |
| Parameters | char \*alp |
| Return Value | None |