Operation System – Assignment\_1 201411303 컴퓨터공학과 이준호

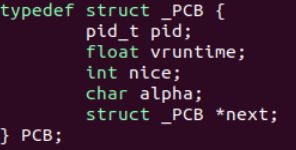
KU\_CFS

1. 목적: 운영체제의 프로세스 스케줄러 구현, 단일 코어를 대신하여 응용프로그램에서 하나의 Ready\_Queue를 vruntime을 기준으로 OS의 CFS와 비슷하게 정렬하고, Signal handler를 이용하여 시간 분배를 해준다.
2. CFS (Completely Fair Process):

Vruntime = vruntime + DeltaExec \* 1.25^nice

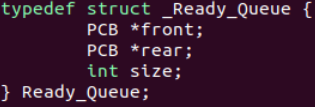
원래의 CFS와 비슷하지만 nice value의 값을 -2, -1, 0, 1, 2로 적은 범위로 계산한다. DeltaExec 값은 최근 Timeslice에서 사용한 시간이지만 이 프로그램에서는 모든 프로세스가 Timeslice를 전부 이용하고 그 Timeslice는 1 second 로 설정한다.

1. 구조체 PCB: 모든 라이브러리는 ku\_cfs.h 해더파일에 넣었다. Process Control Block를 Linked List로 연결하기 위해서 PCB라는 구조체를 선언하였다. PCB 구조체 안에 들어있는 것은 다음과 같다.



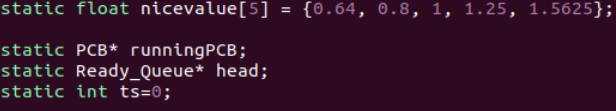
* pid: 프로세스 id
* vruntime: 프로세스의 실행시간을 해당 식에 맞춰 계산하여 넣어준다.
* nice: vruntime을 계산하기 위한 nice value, -2, -1, 0, 1, 2 값 중 하나가 들어간다.
* alpha: 해당 PCB가 갖는 알파벳 값을 넣어준다. 대문자로 넣어준다.
* next: Linked List의 형태 이므로 다음에 해당하는 PCB의 주소를 가진다.

1. 구조체 Ready\_Queue: Ready\_Queue는 다음과 같다.



* front: Ready Queue에 가장 앞에 위치하는 PCB 주소
* rear: Ready Queue에 가장 뒤에 위치하는 PCB 주소
* size: Queue의 크기를 저장한다.

1. 전역변수: Header 파일에 추가적으로 필요한 전역변수들을 선언하였다.



* nicevalue: 1.25^nice에 해당하는 각 값을 -2, -1, 0, 1, 2 순서로 넣었다.
* runningPCB: 현재 실행 중인 PCB를 넣어 사용한다.
* head: Ready Queue를 가르키기 위한 것으로 단일 코어로 가정하였으므로 하나만 설정하였다.
* ts: 전체 Timeslice에 해당하는 값으로 초기값은 0으로 주었지만 main 함수에서 인자값 6번째 값으로 받는 값을 넣어주어 사용한다.

1. main: main함수를 간략히 요약하면 다음과 같다

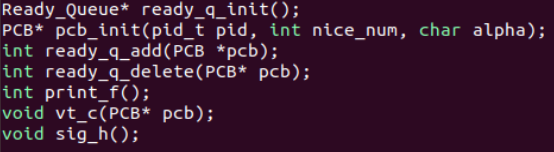
* 프로그램 실행 시 넣어주는 변수 6개를 해당하는 변수에 넣어준다.

n[5]에 -2, -1, 0, 1, 2 순서로 nice값에 따른 프로세스의 개수를 넣는다.

전역변수 ts에 전체 timeslice를 넣는다.

* 각 nice값에 따른 프로세스를 fork해주고 큐에 넣어준다. 이때, execl로 ku\_app을 실행한다.
* sleep(5)로 초기화하는 시간을 넣어준다.
* itimer와 signal을 설정해준다.
* Ready\_Queue의 front에 위치하고 있는 PCB를 큐에서 제거하면서 실행 프로세스로 변경한다. (ts 만큼 반복)
* 마지막으로 실행중인 프로세스의 PCB를 큐에 넣어주고 SIGSTOP해준다.
* SIGKILL Flog와 free를 이용하여 생성했던 프로세스와 PCB를 제거해준다.
* 종료

이외의 다른 함수들 선언이다.



1. Functions

|  |  |  |
| --- | --- | --- |
| Ready\_Queue\* ready\_q\_init( ) | Functionality | 구조체 Ready\_Queue를 초기화 해준다. |
| Parameters | void |
| Return Value | 할당한 Ready\_Queue의 주소를 반환 |

|  |  |  |
| --- | --- | --- |
| PCB\*  pcb\_init  (pid\_t pid,  int nice\_num,  char alpha) | Functionality | 구조체 PCB를 초기화 해준다. |
| Parameters | pid\_t pid: 프로세스의 pid  int nice\_num: nice value  char alpha: 해당 프로세스의 알파벳 |
| Return Value | 할당한 PCB의 주소를 반환 |

|  |  |  |
| --- | --- | --- |
| int  ready\_q\_add (PCB \*pcb) | Functionality | 해당 PCB와 Ready\_Queue에 들어있는 PCB의 vruntime을 비교하여 vruntime이 작은 것이 front에 가깝게 위치하도록 sorting한다. 같을 경우 나중에 들어온 것이 앞에 위치한다. |
| Parameters | PCB\* pcb: Ready\_Queue에 추가하려고 하는 PCB 주소 |
| Return Value | 성공 시 0 반환 |

|  |  |  |
| --- | --- | --- |
| int ready\_q\_delete (PCB\* pcb) | Functionality | 인자 값으로 받는 PCB를 큐에서 삭제한다. |
| Parameters | PCB\* pcb: Ready\_Queue에서 삭제하고자 하는 PCB 주소 |
| Return Value | 성공 시 0 반환 |

|  |  |  |
| --- | --- | --- |
| int  print\_f( ) | Functionality | Ready\_Queue에 있는 모든 PCB를 출력한다. (실제 함수에는 사용하지 않고 확인용으로만 사용하였다.) |
| Parameters | void |
| Return Value | 성공 시 0 반환 |

|  |  |  |
| --- | --- | --- |
| void  vt\_c  (PCB\* pcb) | Functionality | 인자 값으로 받은 PCB의 vruntime을 갱신한다. |
| Parameters | PCB\* pcb: vruntime을 계산하려는 PCB 주소 |
| Return Value | void |

|  |  |  |
| --- | --- | --- |
| void  sig\_h( ) | Functionality | Signal handler Function으로 Ready\_Queue에 가장 앞, 즉 front에 위치하고 있는 PCB를 꺼내 Running 시키고 vruntime값을 갱신한 후 전체 timeslice인 ts를 줄여준다. |
| Parameters | void |
| Return Value | void |