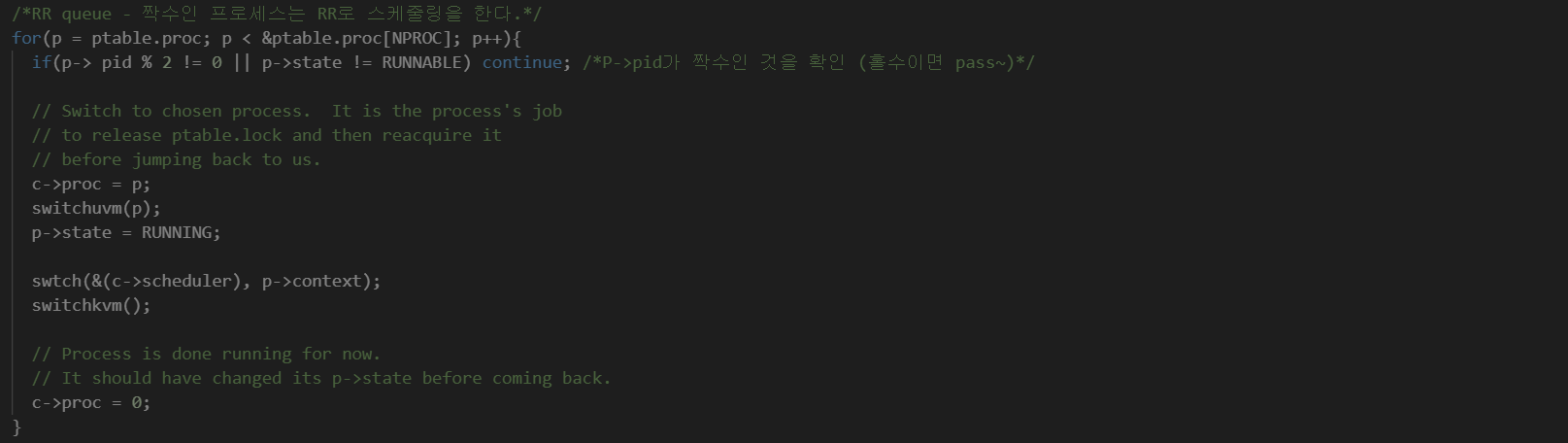
**OS\_Project02\_2017069598 박상지**

|  |  |
| --- | --- |
| **RR scheduler** | 기본적으로 RR 스케줄러를 명세서데로 설정을 하였다. RR 스케줄러는 Default 값으로 설정을 하고 짝수인 pid인 프로세스들을 RR으로 스케줄링을 하였다. |
| **FCFS scheduler** | 홀수 pid를 갖는 프로세스를 FCFS으로 스케줄링을 하였고, sleeping 되어진 프로세스 중에서 pid가 작은 홀수 값을 비교해서 스케줄링을 하였다. |
| **Multilevel queue scheduler** | 우선적으로 짝수 pid를 RR으로 스케줄링을 하였고, 후에 짝수 프로세스를 처리하고 짝수 프로세스가 RR 스케줄링으로 모두 처리되면 후에 sleeping 되어진 홀수 프로세스를 FCFS으로 스케줄링을 하였다. 도중에 짝수 pid가 들어오면 처리하고 있는 홀수 프로세스가 끝이나면 다시 RR으로 스케줄링으로 처리하게끔 설정하였다. |
| **MLFQ scheduler** | 구현 도중 디버깅 오류. |

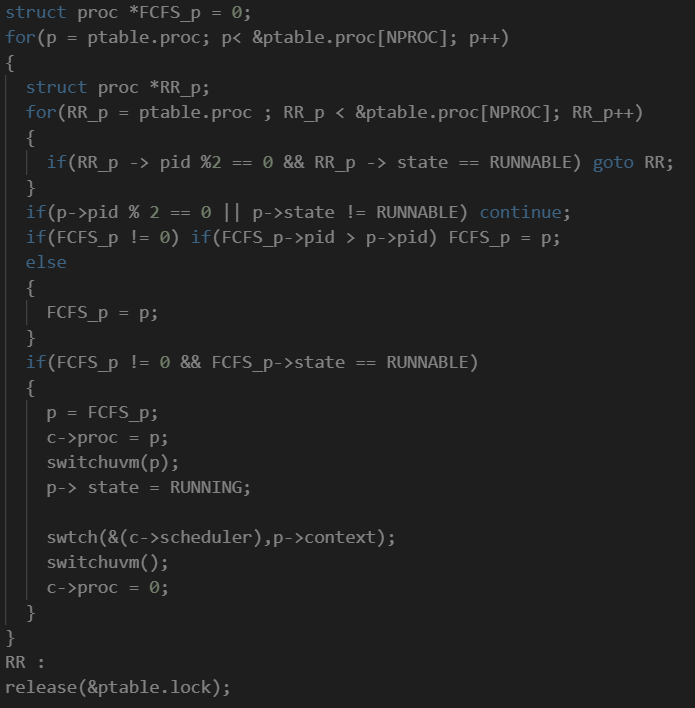
**Multilevel queue scheduler**

**RR scheduler**: 우선적으로 모든 프로세스는 RR 스케줄러로 들어간다. 해당 프로세스의 PID가 홀수인 프로세스는 모두 무시를 한다. 물론 해당 프로세스가 작동하는 상태가 아니여도 무시를 한다. 그 다음 모든 짝수 프로세스가 종료되면 FCFS 스케줄러로 넘어간다. RR 스케줄러는 <그림1>와 같이 구현을 하였다.

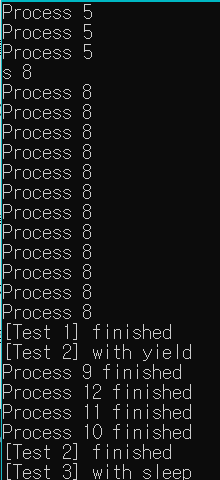
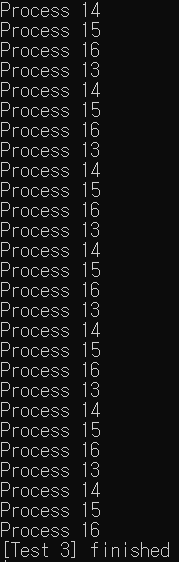
<그림1>

**FCFS scheduler:** FCFS에서는 pid가 2로 나누어 떨어지고 RR 프로세스가 돌아가고 있으면 다시 RR로 넘어가게 설정을 시켜줬다. (즉, RR 프로세스가 FCFS 프로세스보다 우선순위를 높게 설정을 했다.) 현재 FCFS 스케줄러로 돌아가는 pid 보다 작은 pid를 갖고 있는 프로세스를 우선적으로 처리하게끔 설정하였다. FCFS는 <그림2>와 같이 구현을 하였다.

<그림2>



실행 결과:

현재 p2\_ml\_test.c 를 실행시킨 결과이다. test 2를 보면 프로세스가 9가 끝나고 프로세스 12가 실행되었는데 이것은 fcfs 스케줄러로 인해서 프로세스가 끝날 때까지 들어온 프로세스는 대기했다가 들어온 프로세스 중에서 짝수 프로세스를 RR 프로세스로 먼저 실행을 시켜서 우선순위가 성성공적으 된 것을 확인할 수 있다. test3까지 모두 성공적으로 종료되었기 때문에 multilevel test는 올바르게 동작하고 있다는 것을 예상할 수 있다.

트러블 슈팅: multilevel queue를 구현하는데 어려움이 아닌 어려움이 많았는데 모든 트러블은 사소한 것 많이 발생하였다.

초기에 makefile에 default 값을 선언을 해주지 않아 해당 테스트들이 무한루프에 빠지는 상황이 왔었다. default 값을 선언해주지 않았는데 어떻게 make가 되었는지 아직도 의문이다. 그리고 우선 순위를 주지 않아서 프로세스가 순차적으로 진행되는 상황이 왔는데 해당 RR 프로세스가 동작하고 있는 상태면 해당 프로세스를 lock을 걸게 설정해주었다. yield 함수를 각각의 system call에 선언해주지 않아 작은 문제를 발생하기도 했지만 system call에 선언해주고 트러블은 모두 사라졌다.