<Project 4> 2015310884 박소현

1. proc.c에서 void scheduler(void) 함수 수정

priority based scheduler가 되도록 void scheduler(void) 함수를 수정한다. 방법을 설명하자면, 테이블을 돌 때마다 한 번 더 테이블을 돌며 RUNNABLE이면서 우선순위가 더 높은 프로세스를 highP로 넣어준다. 그리고 이 highP가 p(프로세스)로 선정되어 실행되도록 한다. 코드로 표현하면, 파란 부분이 추가된다. 그리고 이 방식은 테이블을 순차대로 돌면서 확인을 한 후 highP에 넣어주는 방식이므로, 같은 순위의 프로세스는 round-robin 방법으로 선택되어 실행된다.

void

scheduler(void)

{

struct proc \*p;

struct proc \*p1;

struct proc \*highP;

for(;;){

...

for(p=ptable.proc; p < &ptable.proc[NPROC]; p++) {

...

//add below code

highP = p;

for(p1 = ptable.proc; p1 < &ptable.proc[NPROC]; p1++) {

if(p1->state != RUNNABLE)

continue;

if(highP->nicevalue > p1->nicevalue)

highP = p1;

}

p = highP;

...

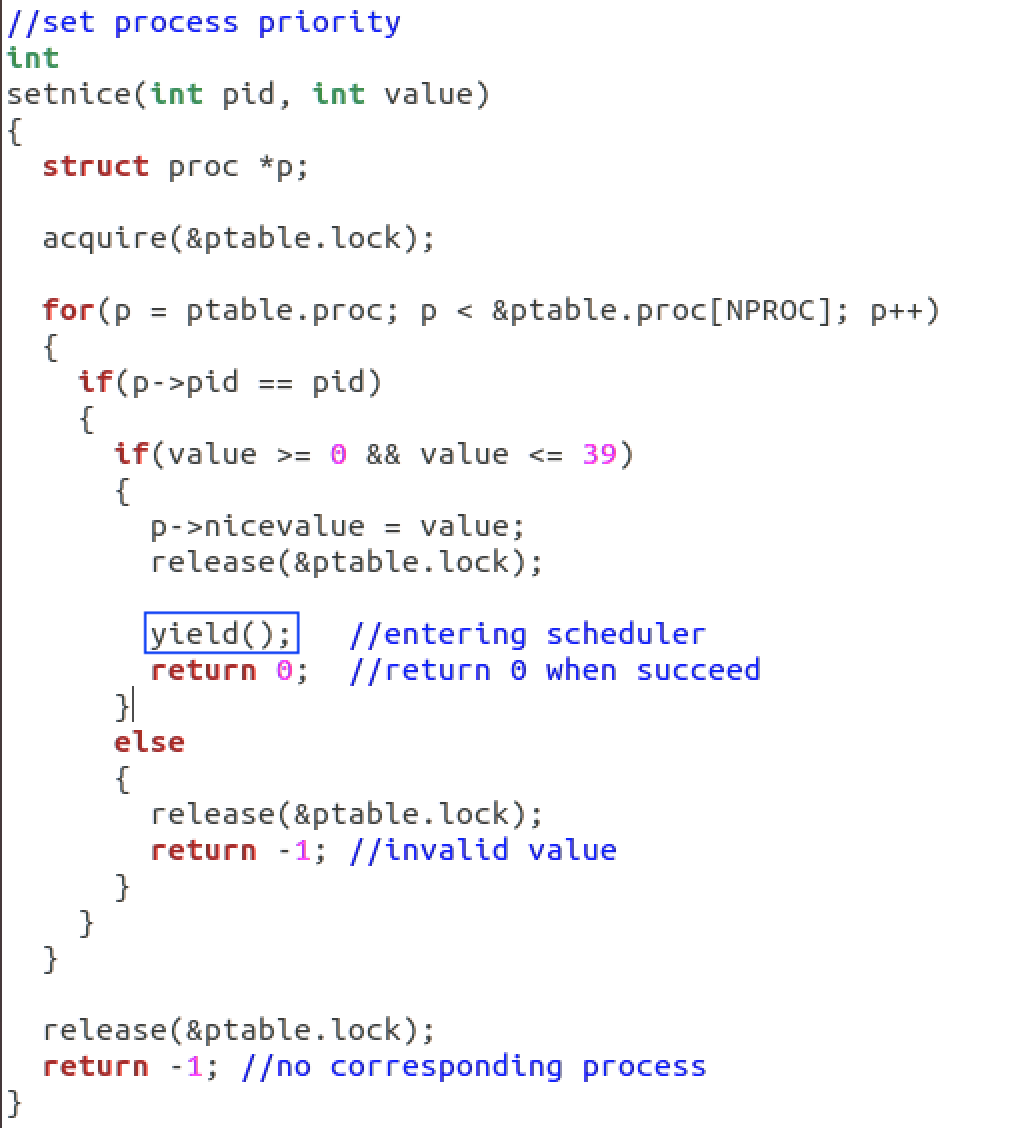
}

}

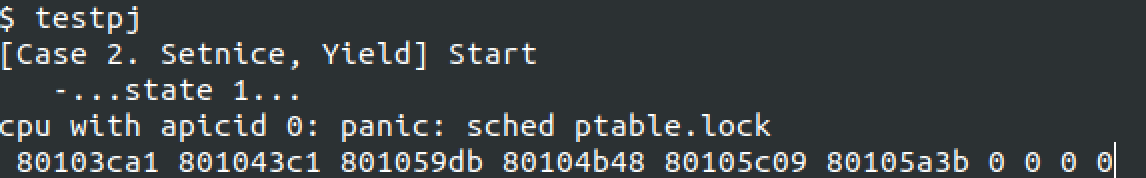
}

2. proc.c 에서 int setnice(int pid, int value) 함수 수정

setnice 함수가 호출될 때도 스케줄러가 호출될 수 있도록 yield();를 다음 위치에 추가한다.



sched();를 쓰면 다음과 같은 에러가 뜬다.



sched();는 ptable.lock을 hold하고 있을 때만 사용이 가능한데, 위 코드 상에서 release가 된 상황이기에 사용이 불가한 것이다. 따라서 sched();대신에 yield();를 쓰도록 한다.