1. 프로젝트 개요
   1. Pintos 커널의 timer\_sleep() 함수를 개선한다.
   2. Pintos 에 우선순위 스케줄러를 구현한다. 또한 우선순위 스케줄링 시 발생할 수 있는 우선순위 역전을 방지할 수 있는 priority donation 기능을 구현한다.
   3. [Extra credit] 다단계 피드백 큐 스케줄러를 구현한다.
2. 프로젝트 내용’
   1. Alarm clock의 개선

* [문제정의] ‘devices/timer.c’에 정의되어 있는 timer\_sleep() 함수를 개선한다. 현재의 timer\_sleep()은 기능적으로는 정상 동작하나, 정해진 시간이 경과할 때까지 thread\_yield()를 반복 호출하며 busy waiting 하도록 구현되어 있다. 이를 busy waiting 없이 수행하도록 수정하여 개선한다.
* [해결] void timer\_sleep (int64 t ticks)를 호출한 쓰레드를 현재 시각 기준으로 ticks 시간이 경과할 때까지 block 시키도록 수정한다. 이후 ticks timer tick 이상이 경과하면 해당 thread를 ready 상태로 전이시키면 된다. 이를 위해 timer interrupt가 발생할 때마다 timer tick이 ticks 이상 지났는지 검사하는 방식으로 구현한다.
* [테스트] 구현된 timer\_sleep() 함수가 제대로 동작하는지 여부는 테스트 alarm-single, alarm-multiple, alarm-simultaneous, alarm-priority, alarm-zero, alarm-negative를 사용하여 시험한다.
  1. 우선순위 스케줄러 구현
* [문제정의] (1) 현재 pintos에는 라운드로빈 방식의 스케줄러가 구현되어 있다. 이를 스레드 별 우선순위에 따라 스케줄링 할 수 있는 우선순위 스케줄러를 새로 구현한다. (2) 우선순위 스케줄링 방식의 문제점은 우선순위 역전 현상이 일어날 수 있다는 것이다. 이를 방지할 수 있도록 우선순위 스케줄러 priority donation 기능을 구현한다.
* [해결] 각 스레드가 자신의 우선순위를 확인하고 우선순위를 변경할 수 있도록 두 가지 함수 void thread\_set\_priority(int new\_priority)와 int thread\_get\_priority(void)를 구현한다.
* [테스트] 구현된 우선순위 스케줄러가 제대로 동작하는지 여부는 테스트 priority-change, priority-preempt, priority-fifo, priority-sema, priority-convar, priority-donate-one, priority-donate-multiple, priority-donate-multiple2, priority-danate-nest, priority-donate-chain, priority-donate-sema, priority-donate-lower를 사용하여 시험한다.