----GROUP----

권민철 <ventania1680@gmail.com>
김준영 <john5910@naver.com>
이장원 <rex94@naver.com>

----PRELIMINARIES----

권민철 : 1
김준영 : 1
이장원 : 1

함께 모여 기존 코드 분석하고, 모르는 부분은 웹페이지 참고하여 수정하였습니다.

----DATA STRUCTURE----

```
struct alarm{
    int64_t wait_time;
    struct thread *th;
    struct list_elem elem;
};
static struct list wait_queue;
```

A1 ->
쓰레드에 대기 시간에 관한 attribute가 없어서 이를 추가했습니다. (int64_t wait_time)
thread에 있는 list_elem을 사용할 수 있지만, 그렇게 하면 호출할 때마다 "alarm->th->elem" 과 같은 형식으로 메모리를 여러 번 참조하기 때문에 alarm 구조체에 list_elem을 추가했습니다.

----ALGORITHM----

```
void
timer_sleep(int64_t ticks)
{
    enum intr_level old_intr_level;

    ASSERT(intr_get_level() == INTR_ON);

    old_intr_level = intr_disable();
    alarm_block((timer_ticks() + ticks), thread_current());
    intr_set_level(old_intr_level);
}

void
alarm_block(int64_t wait_time, struct thread *th)
{
    struct alarm *t = (struct alarm*)malloc(sizeof(struct alarm));
    t->wait_time = wait_time;
    t->th = th;

    list_insert_ordered(&wait_queue, &t->elem, less_compare, (void *)NULL);
    thread_block();
}

void
alarm_unblock(void)
{
    struct alarm *t;
    while(!list_empty(&wait_queue)){
        t = list_entry(list_front(&wait_queue), struct alarm, elem);
        if(t->wait_time <= timer_ticks()){
            list_pop_front(&wait_queue);
            thread_unblock(t->th);
        }
        else break;
    }
}
```

```
static void
timer_interrupt (struct intr_frame *args UNUSED)
{
  ticks++;
  thread_tick ();
  alarm_unblock();
}
```

A2 ->

1. ASSERT함수 사용, interrupt가 활성화 되어있지 않으면 error message 출력
2. intr_disable()을 사용, interrupt level을 저장하고 더 이상 interrupt를 받지 않도록 한다.
3. alarm_block() 호출,
   3.1. 현재 실행중인 thread의 정보와 input으로 받은 대기 시간을 담고 있는 alarm 구조체를 만들고, list에 추가한다.
   3.2. thread_block()을 사용, 현재 실행중인 thread를 block한다.
4. timer_interrupt()가 실행되면서 tick이 증가하고, alarm_unblock()이 호출된다.
   4.1. list에 있는 thread 중 대기 시간이 모두 지난 thread를 찾는다.
   4.2. 대기 시간이 지난 thread는 wait queue에서 pop하고, thread_unblock()을 통해서 unblock한다.

A3 ->

기존 sleep에서는 thread가 ready queue의 끝으로 들어가서 scheduler가 계속 push back을 해줘야 하는 문제가 있는데, wait queue를 만들어서 이 작업을 최소화했습니다.

timer_interrupt()가 alarm_unblock()을 호출합니다.
alarm_block()에서 list_insert_ordered로 wait queue를 대기시간 오름차순으로 정렬하여
alarm_unblock()에서 대기시간이 지난 thread만 ready queue로 보냅니다.