Name: Hemal Paneliya, Anil Pavuluru

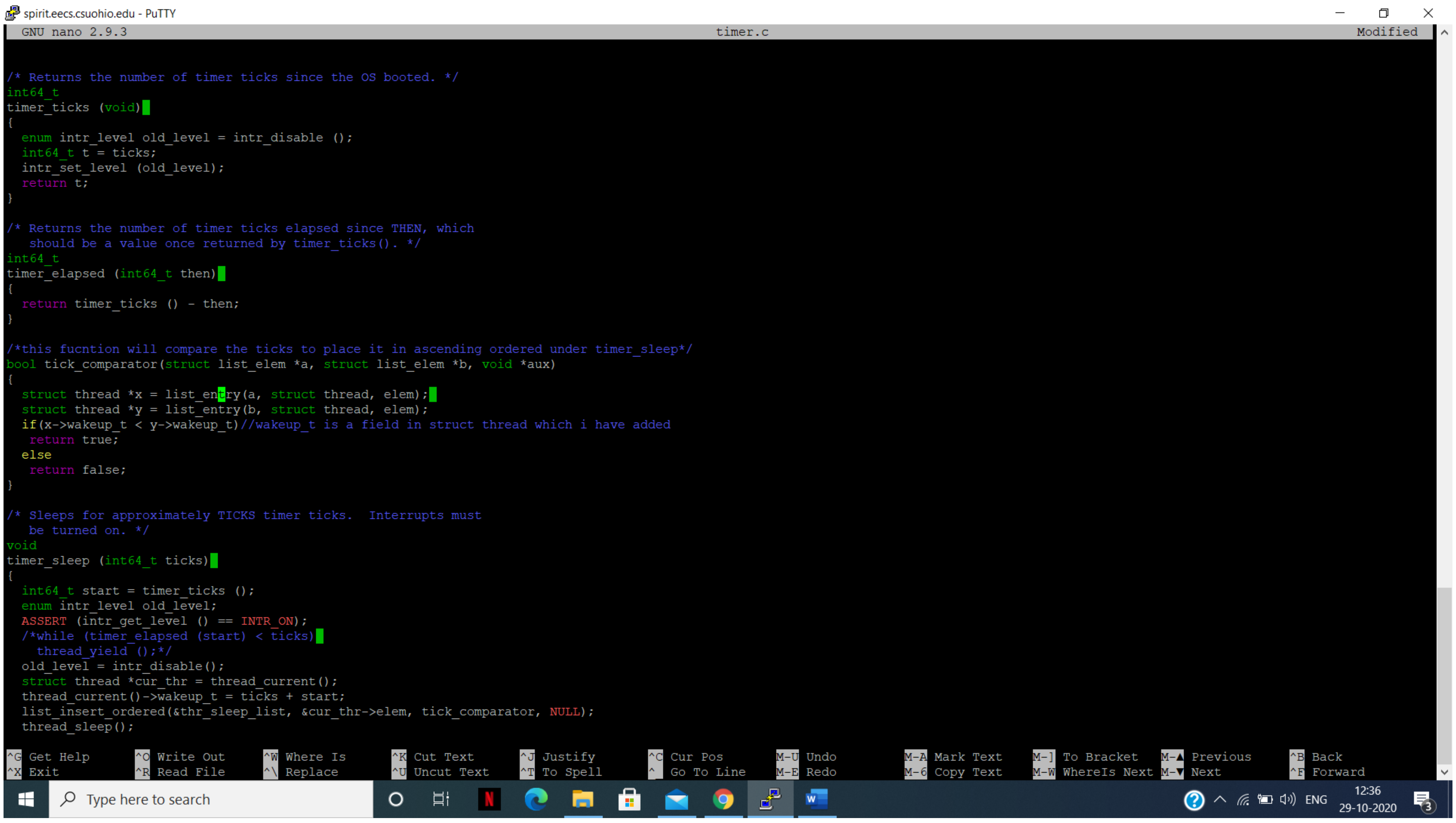
Login id: anpavulu

Project Status: Not Working

Task #1:

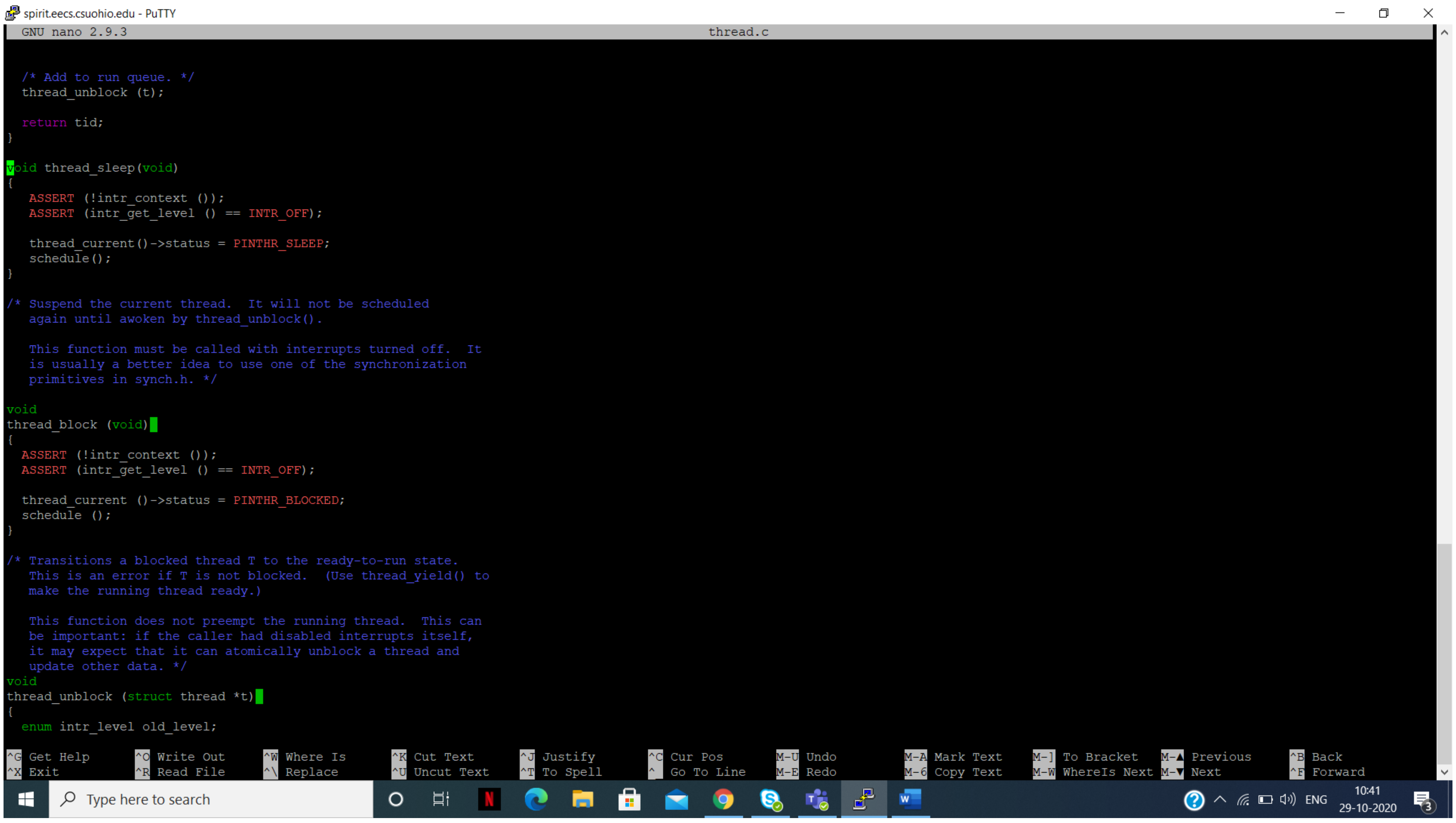
In this task we have modify timer.c to overcome busy yielding. For this we have created an ordered thr_sleep_list and change its state to PINTHR_SLEEP to overcome busy yielding. Instead of calling thread_yield() I cam calling thread_sleep() under thread.c, to make the status of my threads to PINTHR_SLEEP.

Under timer.c :



```
  GNU nano 2.9.3                     timer.c                     Modified


/* Returns the number of timer ticks since the OS booted. */
int64_t
timer_ticks (void)
{
  enum intr_level old_level = intr_disable ();
  int64_t t = ticks;
  intr_set_level (old_level);
  return t;
}

/* Returns the number of timer ticks elapsed since THEN, which
   should be a value once returned by timer_ticks(). */
int64_t
timer_elapsed (int64_t then)
{
  return timer_ticks () - then;
}

/*this fucntion will compare the ticks to place it in ascending ordered under timer_sleep*/
bool tick_comparator(struct list_elem *a, struct list_elem *b, void *aux)
{
  struct thread *x = list_entry(a, struct thread, elem);
  struct thread *y = list_entry(b, struct thread, elem);
  if(x->wakeup_t < y->wakeup_t)//wakeup_t is a field in struct thread which i have added
   return true;
  else
   return false;
}

/* Sleeps for approximately TICKS timer ticks.  Interrupts must
   be turned on. */
void
timer_sleep (int64_t ticks)
{
  int64_t start = timer_ticks ();
  enum intr_level old_level;
  ASSERT (intr_get_level () == INTR_ON);
  /*while (timer_elapsed (start) < ticks)
    thread_yield ();*/
  old_level = intr_disable();
  struct thread *cur_thr = thread_current();
  thread_current()->wakeup_t = ticks + start;
  list_insert_ordered(&thr_sleep_list, &cur_thr->elem, tick_comparator, NULL);
  thread_sleep();
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket  M-▲ Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  M-W WhereIs Next M-▼ Next     ^F Forward
```

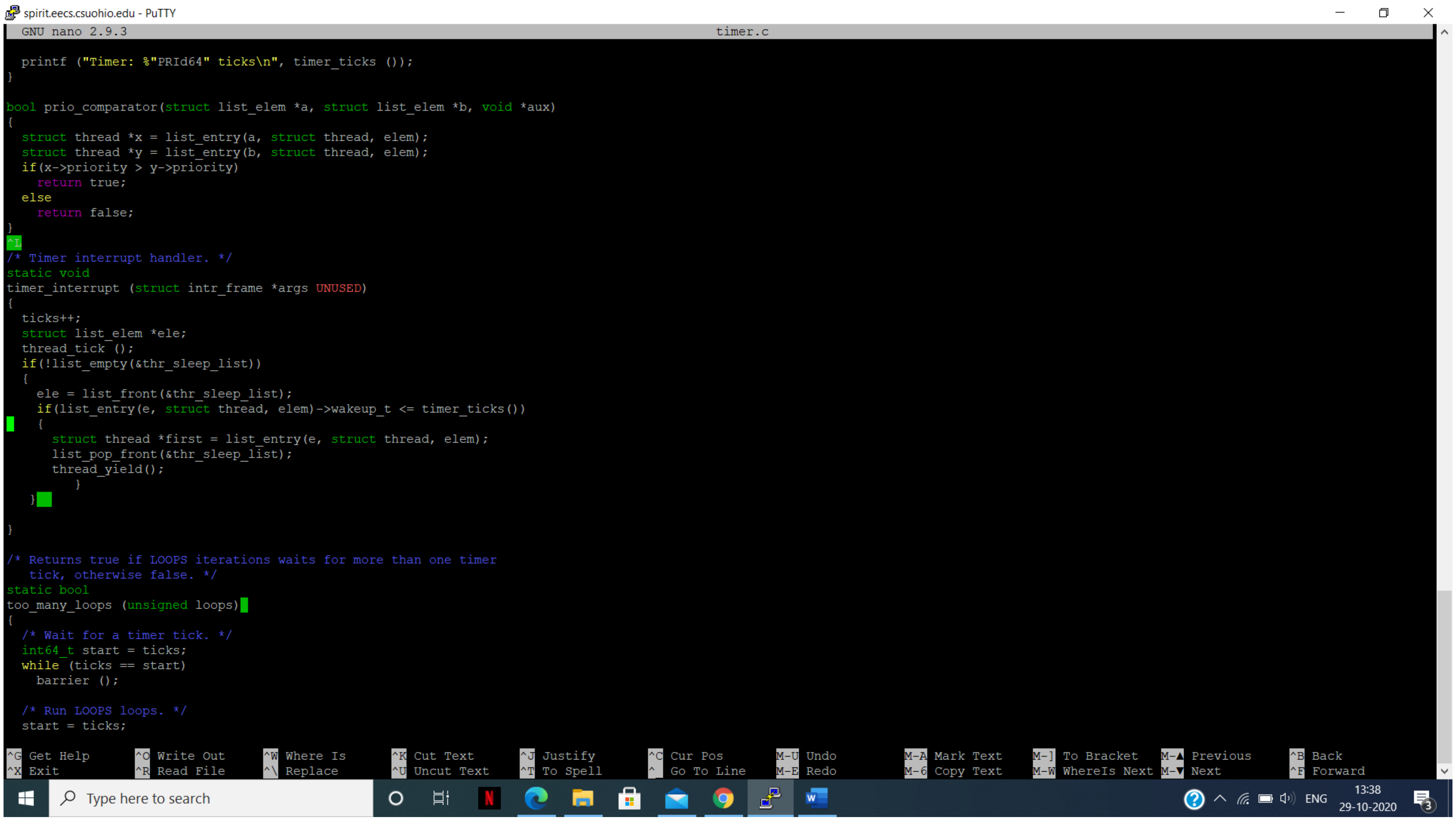
Under thread.c:



```
  GNU nano 2.9.3                     thread.c


  /* Add to run queue. */
  thread_unblock (t);

  return tid;
}

void thread_sleep(void)
{
   ASSERT (!intr_context ());
   ASSERT (intr_get_level () == INTR_OFF);

   thread_current()->status = PINTHR_SLEEP;
   schedule();
}

/* Suspend the current thread.  It will not be scheduled
   again until awoken by thread_unblock().

   This function must be called with interrupts turned off.  It
   is usually a better idea to use one of the synchronization
   primitives in synch.h. */
void
thread_block (void)
{
  ASSERT (!intr_context ());
  ASSERT (intr_get_level () == INTR_OFF);

  thread_current ()->status = PINTHR_BLOCKED;
  schedule ();
}

/* Transitions a blocked thread T to the ready-to-run state.
   This is an error if T is not blocked.  (Use thread_yield() to
   make the running thread ready.)

   This function does not preempt the running thread.  This can
   be important: if the caller had disabled interrupts itself,
   it may expect that it can atomically unblock a thread and
   update other data. */
void
thread_unblock (struct thread *t)
{
  enum intr_level old_level;
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket  M-▲ Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  M-W WhereIs Next M-▼ Next     ^F Forward
```

Task #2:

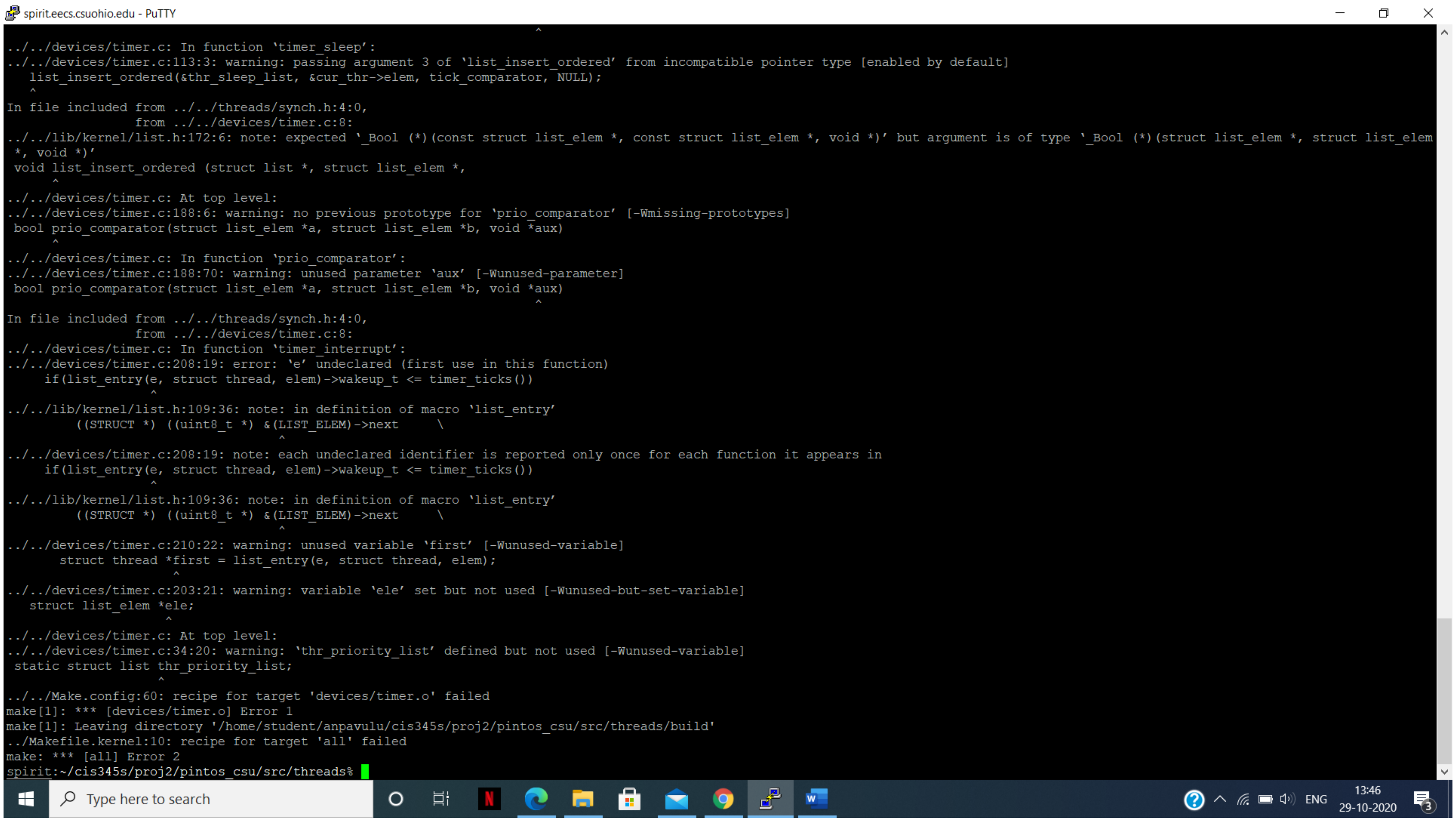
Then to bring these threads to ready_list we have modified timer_interrupt(). In this task we are using thr_priority_list to put the threads in descending order(i.e. higher priority threads first), where we are facing problem and our things are not working . Here my approach is to check whether the thr_sleep_list is empty or not. If it is not empty then we are popping out the first element from the list and trying to placing it to the ready_list.



```
  GNU nano 2.9.3                                                                                      timer.c

  printf ("Timer: %"PRId64" ticks\n", timer_ticks ());
}

bool prio_comparator(struct list_elem *a, struct list_elem *b, void *aux)
{
  struct thread *x = list_entry(a, struct thread, elem);
  struct thread *y = list_entry(b, struct thread, elem);
  if(x->priority > y->priority)
    return true;
  else
    return false;
}
^L
/* Timer interrupt handler. */
static void
timer_interrupt (struct intr_frame *args UNUSED)
{
  ticks++;
  struct list_elem *ele;
  thread_tick ();
  if(!list_empty(&thr_sleep_list))
  {
    ele = list_front(&thr_sleep_list);
    if(list_entry(e, struct thread, elem)->wakeup_t <= timer_ticks())
    {
      struct thread *first = list_entry(e, struct thread, elem);
      list_pop_front(&thr_sleep_list);
      thread_yield();
         }
   }

}

/* Returns true if LOOPS iterations waits for more than one timer
   tick, otherwise false. */
static bool
too_many_loops (unsigned loops)
{
  /* Wait for a timer tick. */
  int64_t start = ticks;
  while (ticks == start)
    barrier ();

  /* Run LOOPS loops. */
  start = ticks;
```

Output:

We are getting this error while building the kernel



```
                                                                    ^
../../devices/timer.c: In function ‘timer_sleep’:
../../devices/timer.c:113:3: warning: passing argument 3 of ‘list_insert_ordered’ from incompatible pointer type [enabled by default]
   list_insert_ordered(&thr_sleep_list, &cur_thr->elem, tick_comparator, NULL);
   ^
In file included from ../../threads/synch.h:4:0,
                 from ../../devices/timer.c:8:
../../lib/kernel/list.h:172:6: note: expected ‘_Bool (*)(const struct list_elem *, const struct list_elem *, void *)’ but argument is of type ‘_Bool (*)(struct list_elem *, struct list_elem
 *, void *)’
 void list_insert_ordered (struct list *, struct list_elem *,
      ^
../../devices/timer.c: At top level:
../../devices/timer.c:188:6: warning: no previous prototype for ‘prio_comparator’ [-Wmissing-prototypes]
 bool prio_comparator(struct list_elem *a, struct list_elem *b, void *aux)
      ^
../../devices/timer.c: In function ‘prio_comparator’:
../../devices/timer.c:188:70: warning: unused parameter ‘aux’ [-Wunused-parameter]
 bool prio_comparator(struct list_elem *a, struct list_elem *b, void *aux)
                                                                      ^
In file included from ../../threads/synch.h:4:0,
                 from ../../devices/timer.c:8:
../../devices/timer.c: In function ‘timer_interrupt’:
../../devices/timer.c:208:19: error: ‘e’ undeclared (first use in this function)
     if(list_entry(e, struct thread, elem)->wakeup_t <= timer_ticks())
                   ^
../../lib/kernel/list.h:109:36: note: in definition of macro ‘list_entry’
         ((STRUCT *) ((uint8_t *) &(LIST_ELEM)->next     \
                                    ^
../../devices/timer.c:208:19: note: each undeclared identifier is reported only once for each function it appears in
     if(list_entry(e, struct thread, elem)->wakeup_t <= timer_ticks())
                   ^
../../lib/kernel/list.h:109:36: note: in definition of macro ‘list_entry’
         ((STRUCT *) ((uint8_t *) &(LIST_ELEM)->next     \
                                    ^
../../devices/timer.c:210:22: warning: unused variable ‘first’ [-Wunused-variable]
       struct thread *first = list_entry(e, struct thread, elem);
                      ^
../../devices/timer.c:203:21: warning: variable ‘ele’ set but not used [-Wunused-but-set-variable]
   struct list_elem *ele;
                     ^
../../devices/timer.c: At top level:
../../devices/timer.c:34:20: warning: ‘thr_priority_list’ defined but not used [-Wunused-variable]
 static struct list thr_priority_list;
                    ^
../../Make.config:60: recipe for target 'devices/timer.o' failed
make[1]: *** [devices/timer.o] Error 1
make[1]: Leaving directory '/home/student/anpavulu/cis345s/proj2/pintos_csu/src/threads/build'
../Makefile.kernel:10: recipe for target 'all' failed
make: *** [all] Error 2
spirit:~/cis345s/proj2/pintos_csu/src/threads%
```