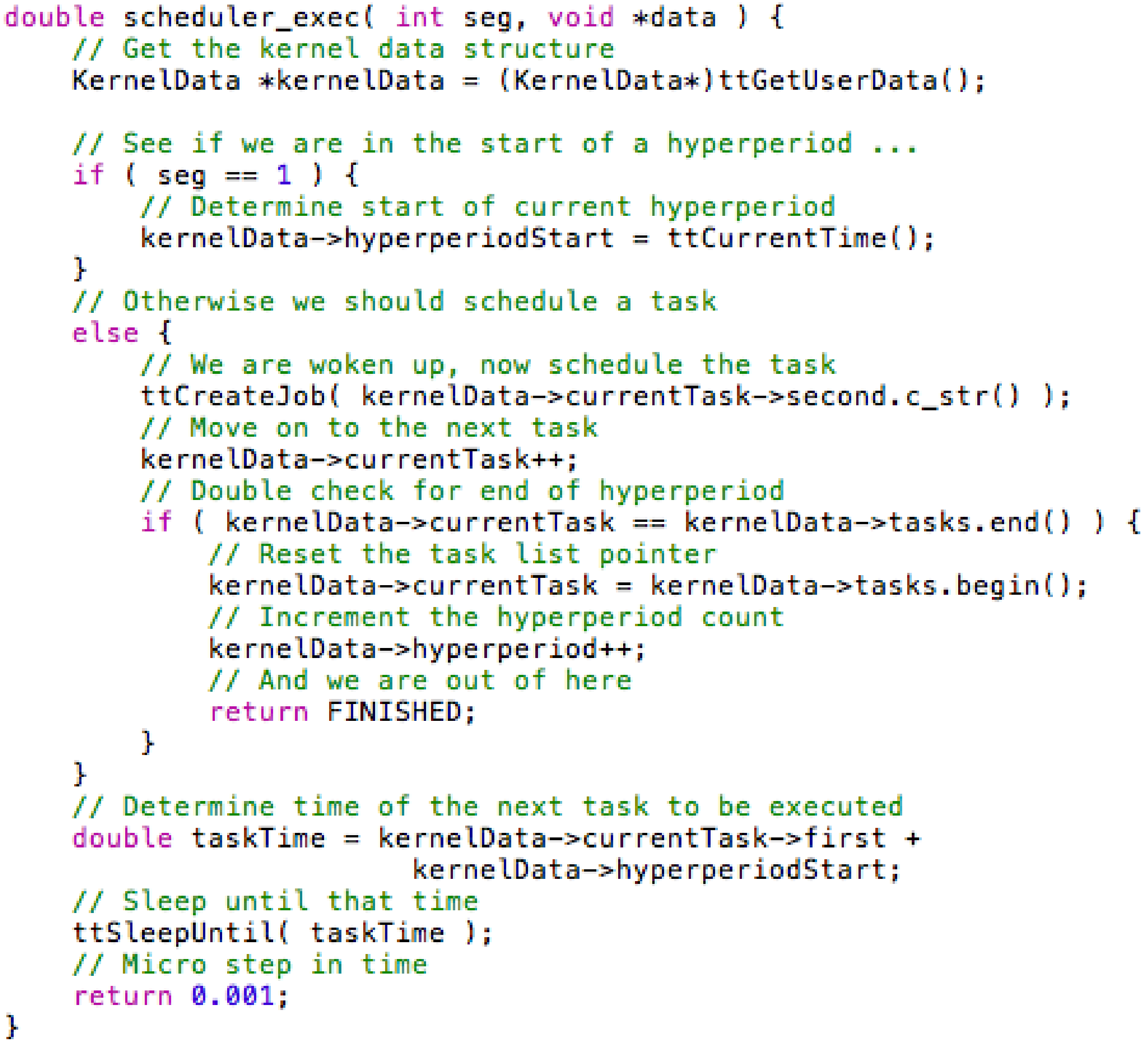

```
double scheduler_exec( int seg, void *data ) {
    // Get the kernel data structure
    KernelData *kernelData = (KernelData*)ttGetUserData();

    // See if we are in the start of a hyperperiod ...
    if ( seg == 1 ) {
        // Determine start of current hyperperiod
        kernelData->hyperperiodStart = ttCurrentTime();
    }
    // Otherwise we should schedule a task
    else {
        // We are woken up, now schedule the task
        ttCreateJob( kernelData->currentTask->second.c_str() );
        // Move on to the next task
        kernelData->currentTask++;
        // Double check for end of hyperperiod
        if ( kernelData->currentTask == kernelData->tasks.end() ) {
            // Reset the task list pointer
            kernelData->currentTask = kernelData->tasks.begin();
            // Increment the hyperperiod count
            kernelData->hyperperiod++;
            // And we are out of here
            return FINISHED;
        }
    }
    // Determine time of the next task to be executed
    double taskTime = kernelData->currentTask->first +
                        kernelData->hyperperiodStart;
    // Sleep until that time
    ttSleepUntil( taskTime );
    // Micro step in time
    return 0.001;
}
```