```
1  #ifndef  __KERNELCORE_H
2
3  #include <stdint.h>
4  #include <stdio.h>
5  #include <LPC17xx.h>
6  #include <stdbool.h>
7
8  #define SHPR3 *(uint32_t*)0xE000ED20 //location of the PendSV priority register
9  #define SHPR2 *(uint32_t*)0xE000ED1C // location of SVC priority register
10 #define ICSR *(uint32_t*)0xE000ED04 //location of the ISCR
11
12 void kernelInit(void); //initialize memory structures and interrupts necessary to run the kernel
13
14 bool osKernelStart(); //start running the kernel, i.e. the OS
15
16 void osCreateMutex(); //create a new mutex in the mutex struct array
17
18 void osAcquireMutex(int mutexID); //determine if the thread is allowed to run otherwise block
19
20 void osReleaseMutex(int mutexID); //make mutex available and/or give it to the next blocked thread
21
22 void osLoadFirst(); //called by the kernel to start running the very first thread, before getting into
   context switching
23
24 void osYield(void); //called by the kernel to schedule the next thread to run and call the context
   switcher
25
26 void SysTick_Handler(void); //called when thread's timeslice is up, calls scheduler to decide which
   thread to run next
27
28 void scheduler(void); //decides which thread to run next, based off round-robin logic
29
30 void SVC_Handler_Main(uint32_t *svc_args);
31
32 int task_switch(void); //called by the PendSV interrupt to set PSP to the next thread scheduled to run
33
34
35
36 #endif
```