```c
#ifndef  __KERNELCORE_H

#include <stdint.h>
#include <stdio.h>
#include <LPC17xx.h>
#include <stdbool.h>

#define SHPR3 *(uint32_t*)0xE000ED20 //location of the PendSV priority register
#define SHPR2 *(uint32_t*)0xE000ED1C // location of SVC priority register
#define ICSR *(uint32_t*)0xE000ED04 //location of the ISCR

void kernelInit(void); //initialize memory structures and interrupts necessary to run the kernel

bool osKernelStart(); //start running the kernel, i.e. the OS

void osLoadFirst(); //called by the kernel to start running the very first thread, before getting into context switching

void osYield(void); //called by the kernel to signal for the next thread to be scheduled

void osSleep(int sleepTime); //called by kernel to set the non-periodic thread to sleep and signal for the next thread to be scheduled

void SysTick_Handler(void); //interrupt called every ~1ms to decrement timers and potentially pre-empt the running thread

void scheduler(void); //decides which thread to run next, based off EDF logic

void SVC_Handler_Main(uint32_t *svc_args); //interrupt containing yield code

int task_switch(void); //called by the PendSV interrupt to set PSP to the next thread scheduled to run



#endif
```