Excercise 4
Simple real time operating system (RTOS)

Use a simple RTOS to create a digital clock, that displays (hour:min:sec) on the LCD. The keys should perform the following actions:

T0 => increase the hours by 1
T1 => increase the minutes by 1
T2 => stop the clock (while the key is pressed)
T3 => reset the time to 00:00:00

Simple real time operating system (rtos.c, rtos.h, rtos_tasks.c, rtos_tasks.h)

void sch_on(unsigned int slice);
/* RTOS initialization function: enables timer0 interrupts to trigger every *slice* mikroseconds to run the scheduler interrupt service routine *sch_int()/**

void sch_int();
/* task scheduler: executed at timer0 interrupts. It manages the timer match register and runs the task. If the task is too long (longer than the time slice), it traps the system in an endless loop.
***/**

voidfuncptr sch_tab[] = {task1, task2, task3,…};
/* An array of task function pointers. Tasks are executed by the scheduler one at a time in a round robin style. */

Limitations:

1. **All time slices are of fixed lengths**
   - **Set by the *slice* parameter when calling *sch_on()***

2. **All tasks must be completed within one time slice. If a task is longer than the time slice, the scheduler is trapped in an endless loop**

3. **The scheduler contains a fixed number of tasks**
   - **Task array *sch_tab* does not change with time**
   - **All tasks are executed periodicaly: dt = num. tasks * slice**

4. **There are no interrupts**
   - **This simple RTOS does not know asynchronous tasks. Only synchronous tasks are allowed and are called by the scheduler. The only allowed interrupt is the timer0, used by RTOS.**

Digital clock:

```
/* Global seconds counter*/
int time=0;

/* The main task should increment the time counter once
per second */
void mainCounter(){
...
}

/* This tasks converts the seconds counter into
hour:min:sec format for LCD*/
void prepareOutput (){
...
}
```

```
/* The lcd_driver task refreshes the LCD display */
char lcd_buff[33] ;
char *lcd_string=lcd_buff;
void lcd_driver(){
...
}
```

Q: How do we achieve accurate time counting (1 second)?
Q: Where do we check the buttons?