EXPERIMENT 12

NAME: Rahil Sharma
PRN: 18070123062
BATCH: EA-3

SUBJECT: ESRTOS

AIM: To study and utilize the Task Scheduler function in Real Time Operating System.

Theory: An algorithm the RTOS uses to schedule threads. The RTOS scheduler determines which thread to run on each CPU in the system at any given time. Run queue—The queue of threads ready to be scheduled. Threads in the run queue are either executing or ready to execute.

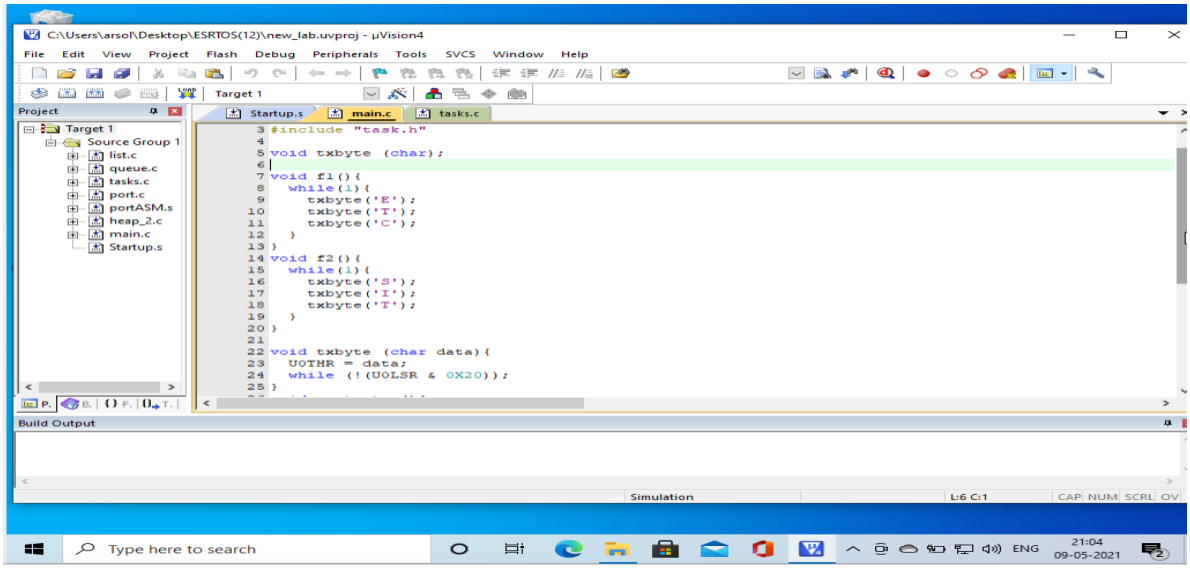
The scheduler is the part of the kernel responsible for deciding which task should be executing at any particular time. The kernel can suspend and later resume a task many times during the task lifetime. The scheduling policy is the algorithm used by the scheduler to decide which task to execute at any point in time.

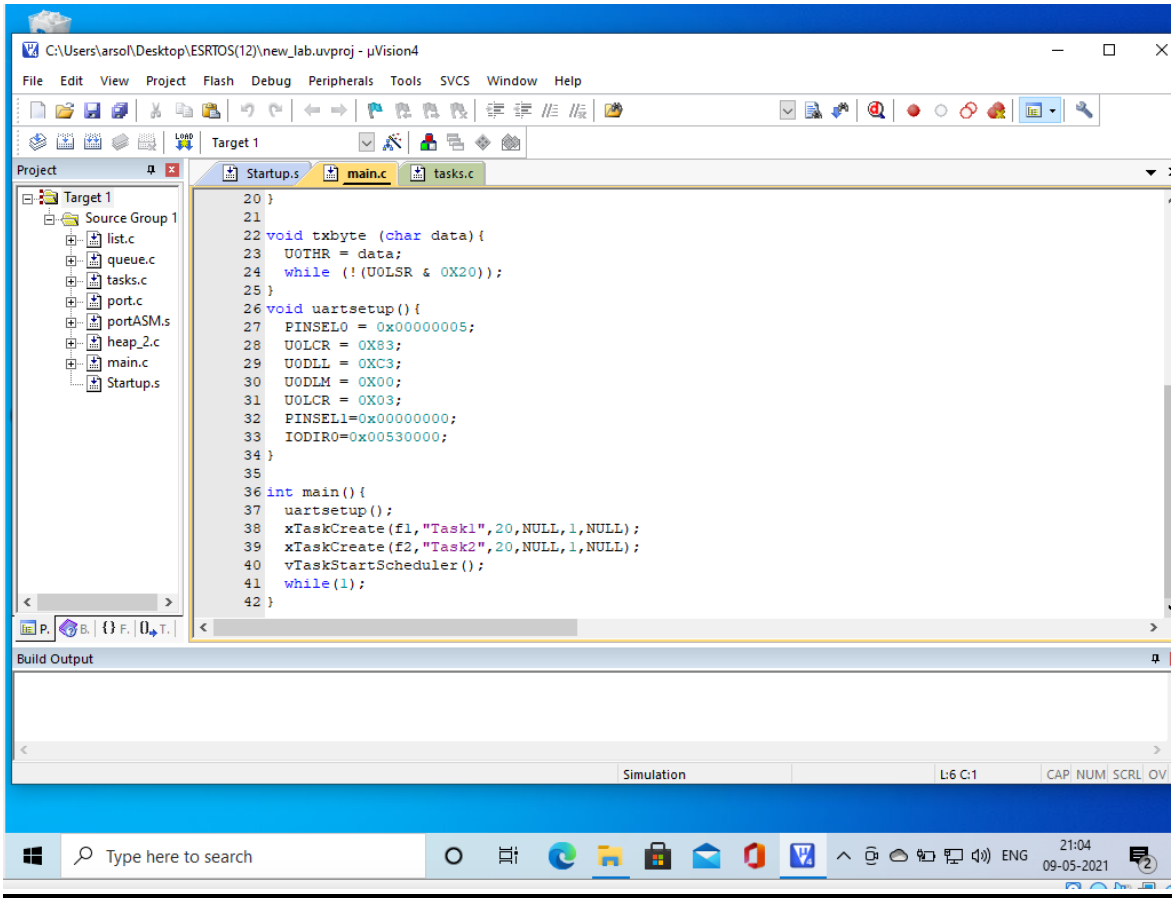
Code of the Program:

```
#include<lpc21xx.h>
#include "FreeRTOS.h"
#include "task.h"
void txbyte (char);
void f1(){
while(1){
txbyte('E');
```

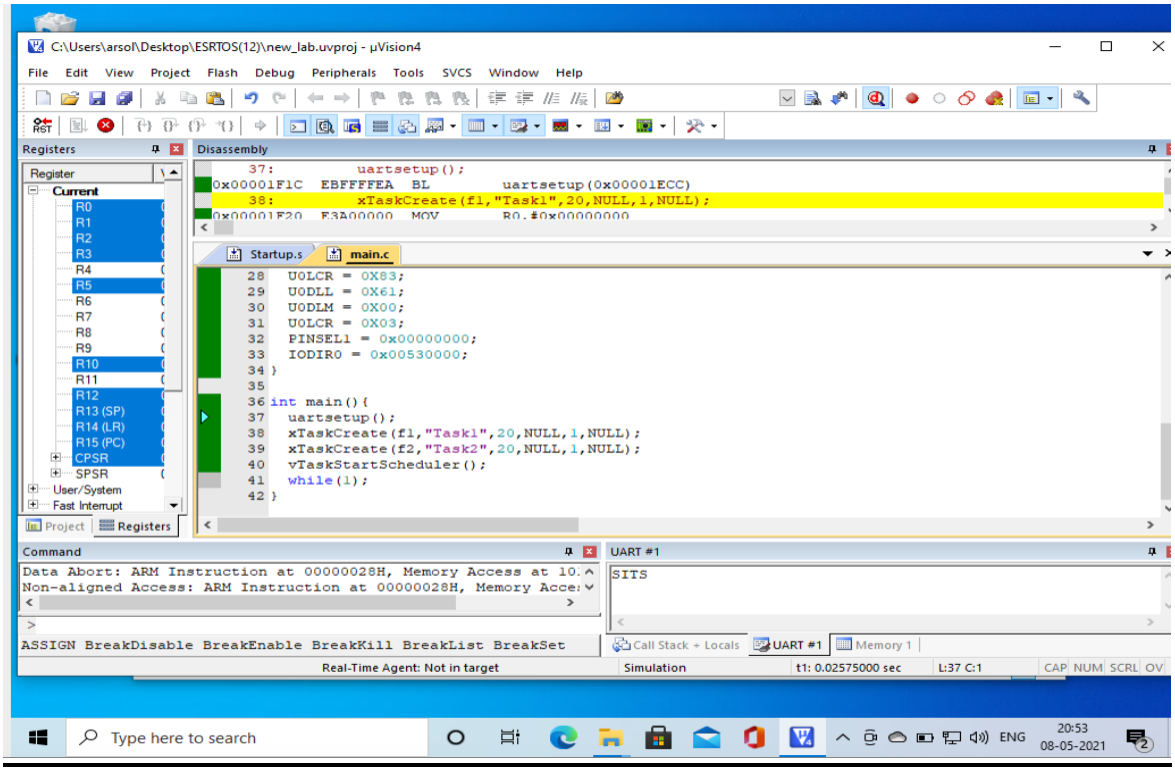
```
txbyte('C');
txbyte('E');
void f2(){
while(1) {
txbyte('S');
txbyte('i');
txbyte('T');
}
}
void txbyte(char data){
U0THR =data;
while (!(U0LSR & 0x20));
}
void uart_setup() {
PINSEL0 = 0X00000005;
U0LCR = 0X83;
U0DLL = 0XC3;
U0DLM = 0X00;
V0LCR = 0X03;
PINSEL1 = 0X00000000;
I0DIRO = 0X00530000;
}
int main() {
 uart_setup();
xTaskCreate(f1,"Task1", 20,NULL,1,NULL);
 xTaskCreate(f2,"Task2",20,NULL,1,NULL);
vTaskStartScheduler();
while(1);
}
```

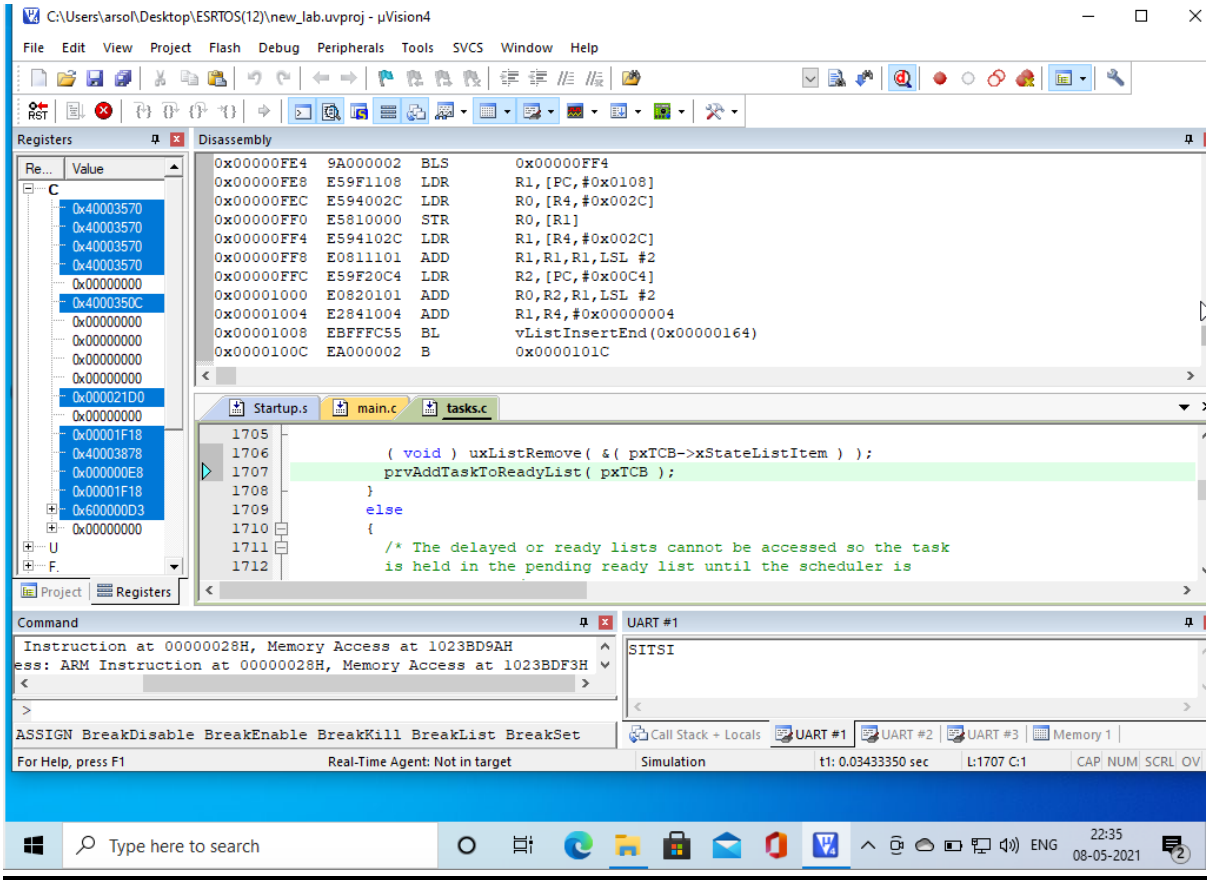
Screenshot of the Program:





Screenshot of the Output:





CONCLUSION: From this experiment we have understood the functioning of Task Scheduler function in Real Time Operating System.