Development of Real-Time Systems
Assignment 2

12/26/2020

Mohamed Ibrahim Elsawy

In This Assignment we have Two Task:

- Matrix task
- Communication task

The Problem

Communication task must send a simulated data packet every 200ms but is often blocked by matrix task.

The Requirements

Create a new task "prioritysettask" which:

- Sets the priority of "communicationtask" to 4 in case its execution time is more than 1000 milliseconds.
- Sets the priority of "communicationtask" to 2 in case its execution time is less than 200 milliseconds.

Code Solution

First declare global variables:

```
//Task Handles
xTaskHandle matrix_handle;
xTaskHandle communication_handle;
TaskHandle_t priority_handle;

//store communication Task execution Time
volatile unsigned long communication_execution_time = 0;
//store matrix Task execution Time
volatile unsigned long matrix_execution_time = 0;
//release time of matrix task Measured from count of ticks since
vTaskStartScheduler was called.
volatile unsigned long matrix_release_time = 0;

//to count only execution time when tasks are really running we
need the following variables.
volatile boolean matrix_active = FALSE;
volatile boolean communication_active = FALSE;
```

The vApplicationTickHook() function will be called by each tick interrupt, so will use it to update communication_execution_time and matrix_execution_time as following:

```
void vApplicationTickHook(void)
{

	// here One Tick == One Milli-Second
	if (communication_active) {
		//increase communication task execution Time Counter
		++communication_execution_time;
	}
	if (matrix_active) {
		//increase matrix task execution Time Counter
		++matrix_execution_time;
	}
}
```

Create new task "prioritysettask":

```
static void prioritysettask() {
	while (1)
	{
		//Sets the priority of "communicationtask" to 4 in case its
execution time is more than 1000 milliseconds .
		if (communication_execution_time > 1000) {
			if (uxTaskPriorityGet(communication_handle) != 4) {
				vTaskPrioritySet(communication_handle, 4);
				printf("change communication task priority to
4. \n");
			}
		}

		//Sets the priority of "communicationtask" to 2 in case its
execution time is less than 200 milliseconds.
		if (communication_execution_time < 200) {
			if (uxTaskPriorityGet(communication_handle) != 2) {
				vTaskPrioritySet(communication_handle, 2);
				printf("change communication task priority to
2. \n");
			}
		}

		// -- Block the task for some ticks before we loop --
		vTaskDelay(1000);
	}
}
```

Create the task in FreeRTOS with the task creation call and give it the highest priority (5):

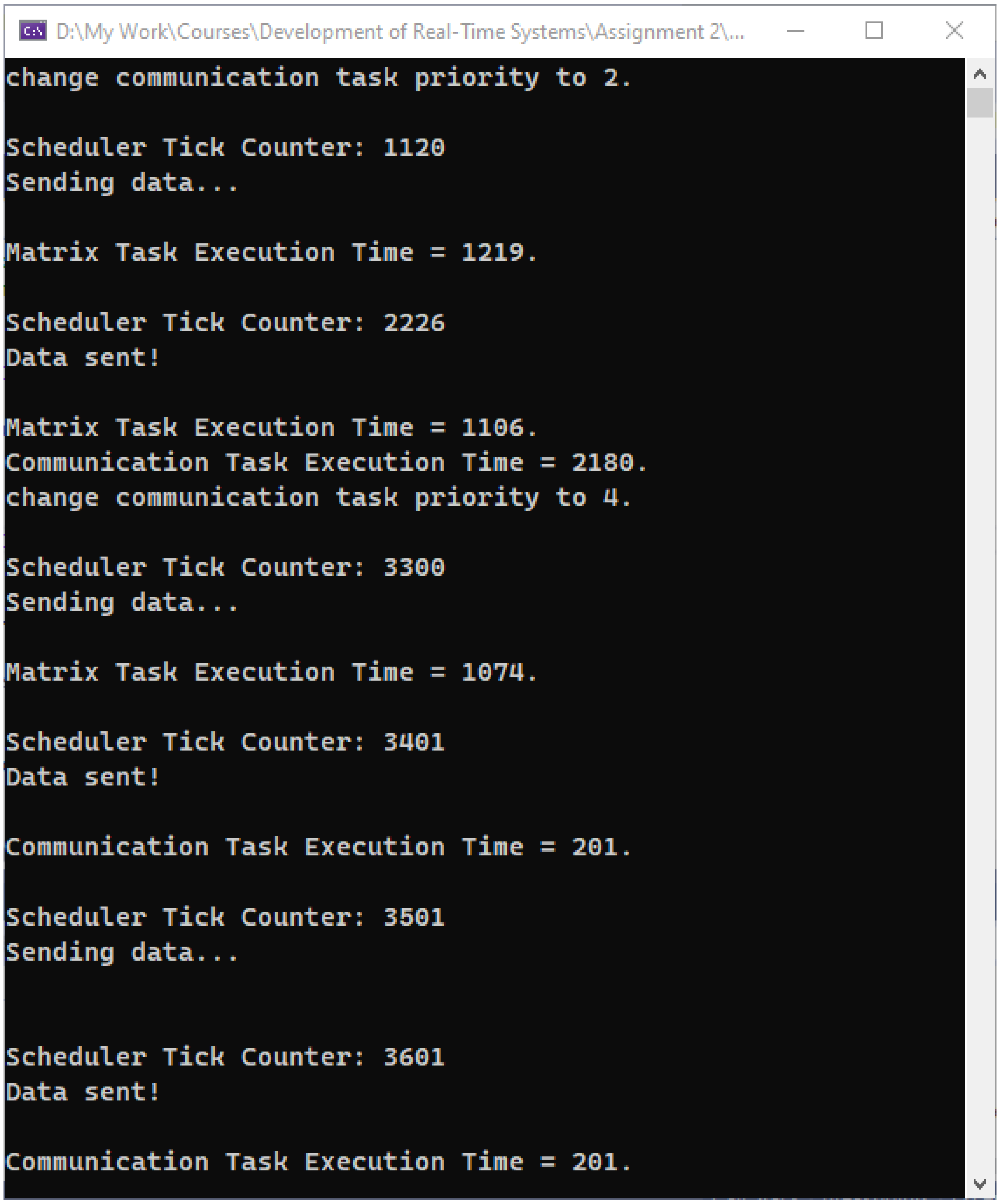
```
xTaskCreate((pdTASK_CODE)prioritysettask, (signed
char*)"Priority", configMINIMAL_STACK_SIZE, NULL, 5,
&priority_handle);
```

Now task "prioritysettask" has the highest priority and period of 1000 ms , that seems Ok but "prioritysettask" will preempt the Task "communicationtask" in middle of its execution time and this will make "prioritysettask" function to decrease "communicationtask" priority to 2. To solve this problem "prioritysettask" task must be suspended at the beginning of "communicationtask" and resume at the end of "communicationtask":

```
static void communication_task()
{
	while (1) {
		// To begin we suspend the priority handle so that it
does not interfere
		vTaskSuspend(priority_handle);
		communication_execution_time = 0;
		communication_active = TRUE;
		printf("\nScheduler Tick Counter: %d \n",
xTaskGetTickCount());
		printf("Sending data...\n\n");
		fflush(stdout);
		vTaskDelay(100);
		printf("\nScheduler Tick Counter: %d \n",
xTaskGetTickCount());
		printf("Data sent!\n\n");
		fflush(stdout);
		vTaskDelay(100);
		printf("Communication Task Execution Time = %d. \n",
communication_execution_time);
		fflush(stdout);
		communication_active = FALSE;
		// After execution we resume the priority handle
		vTaskResume(priority_handle);
	}
}
```

The Result

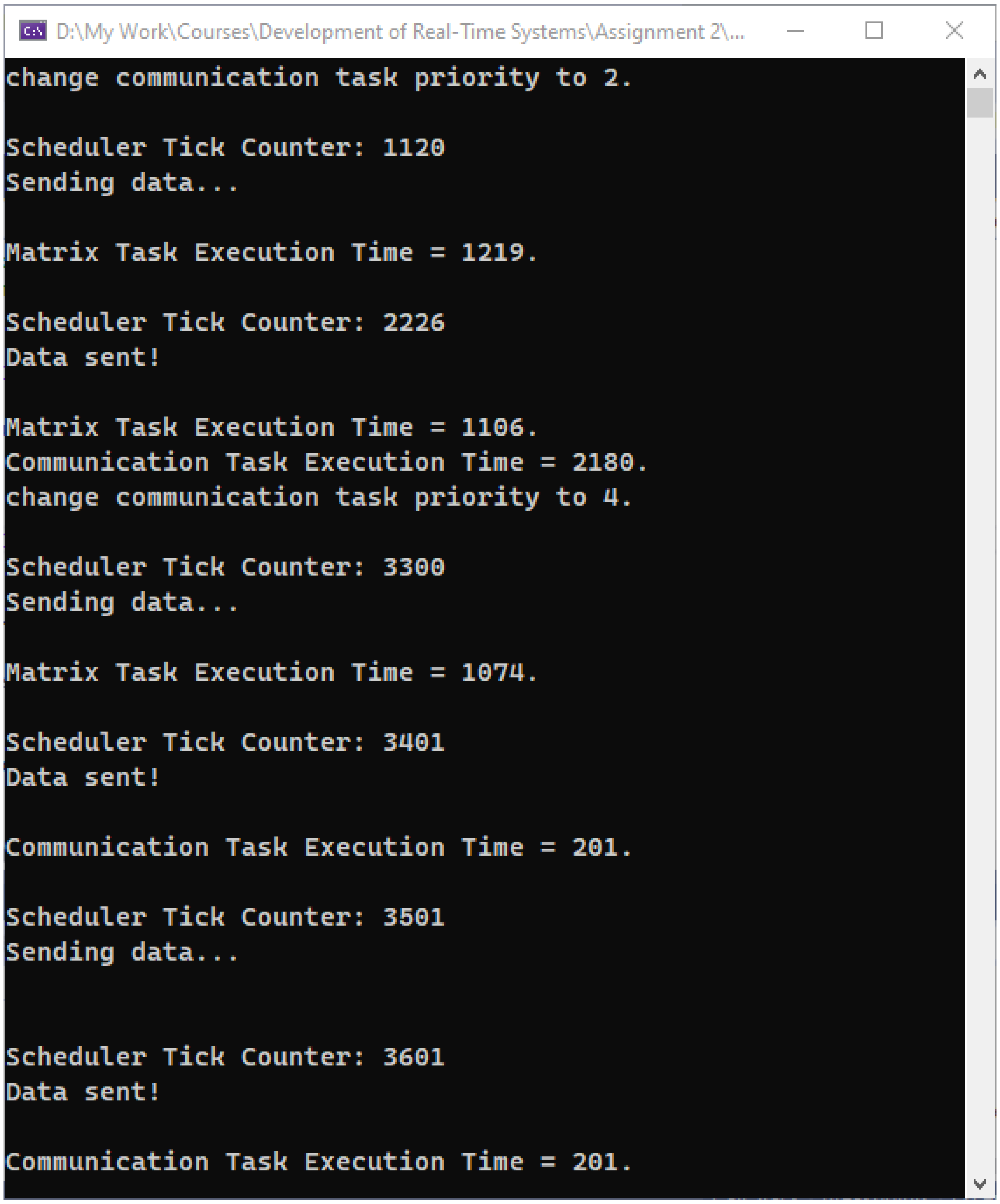
- Before write Code to suspend "prioritysettask" during the execution of "communicationtask". in the fifth time after "communicationtask" priority increase , "prioritysettask" preempt the task and decrease its priority this cause large execution.



D:\My Work\Courses\Development of Real-Time Systems\Assignment 2\...

```
change communication task priority to 2.

Scheduler Tick Counter: 1120
Sending data...

Matrix Task Execution Time = 1219.

Scheduler Tick Counter: 2226
Data sent!

Matrix Task Execution Time = 1106.
Communication Task Execution Time = 2180.
change communication task priority to 4.

Scheduler Tick Counter: 3300
Sending data...

Matrix Task Execution Time = 1074.

Scheduler Tick Counter: 3401
Data sent!

Communication Task Execution Time = 201.

Scheduler Tick Counter: 3501
Sending data...


Scheduler Tick Counter: 3601
Data sent!

Communication Task Execution Time = 201.
```

- After write Code to suspend "prioritysettask" during the execution of "communicationtask". "communicationtask" priority never decrease to 2 because the task execution time large than 200 milli seconds.



D:\My Work\Courses\Development of Real-Time Systems\Assignment 2\...

```
change communication task priority to 2.

Scheduler Tick Counter: 1120
Sending data...

Matrix Task Execution Time = 1219.

Scheduler Tick Counter: 2226
Data sent!

Matrix Task Execution Time = 1106.
Communication Task Execution Time = 2180.
change communication task priority to 4.

Scheduler Tick Counter: 3300
Sending data...

Matrix Task Execution Time = 1074.

Scheduler Tick Counter: 3401
Data sent!

Communication Task Execution Time = 201.

Scheduler Tick Counter: 3501
Sending data...


Scheduler Tick Counter: 3601
Data sent!

Communication Task Execution Time = 201.
```

Why is "matrixtask" using most of the CPU utilization?

freeRTOS use Fixed Priority Scheduler so the matrix task which has higher priority(3) than communication task which has priority of 1 will always be selected to run and matrix multiplication would block the CPU for a long time.

Why must the priority of "communicationtask" increase in order for it to work properly?

In case matrix task has higher priority than communication task, The communication task will only run when matrix task be blocked by `vTaskDelay(100);` at the end of Task's while block. The priority of communication task must be higher than matrix task because communication task must send a simulated data packet every 200ms.

What happens to the completion time of "matrixtask" when the priority of "communicationtask" is increased?

the completion time of matrix task is Noticeably increased because communication task will always preempt matrix task and matrix task will resume its Computation Task after the completion of communication task.

How many seconds is the period of "matrixtask"?

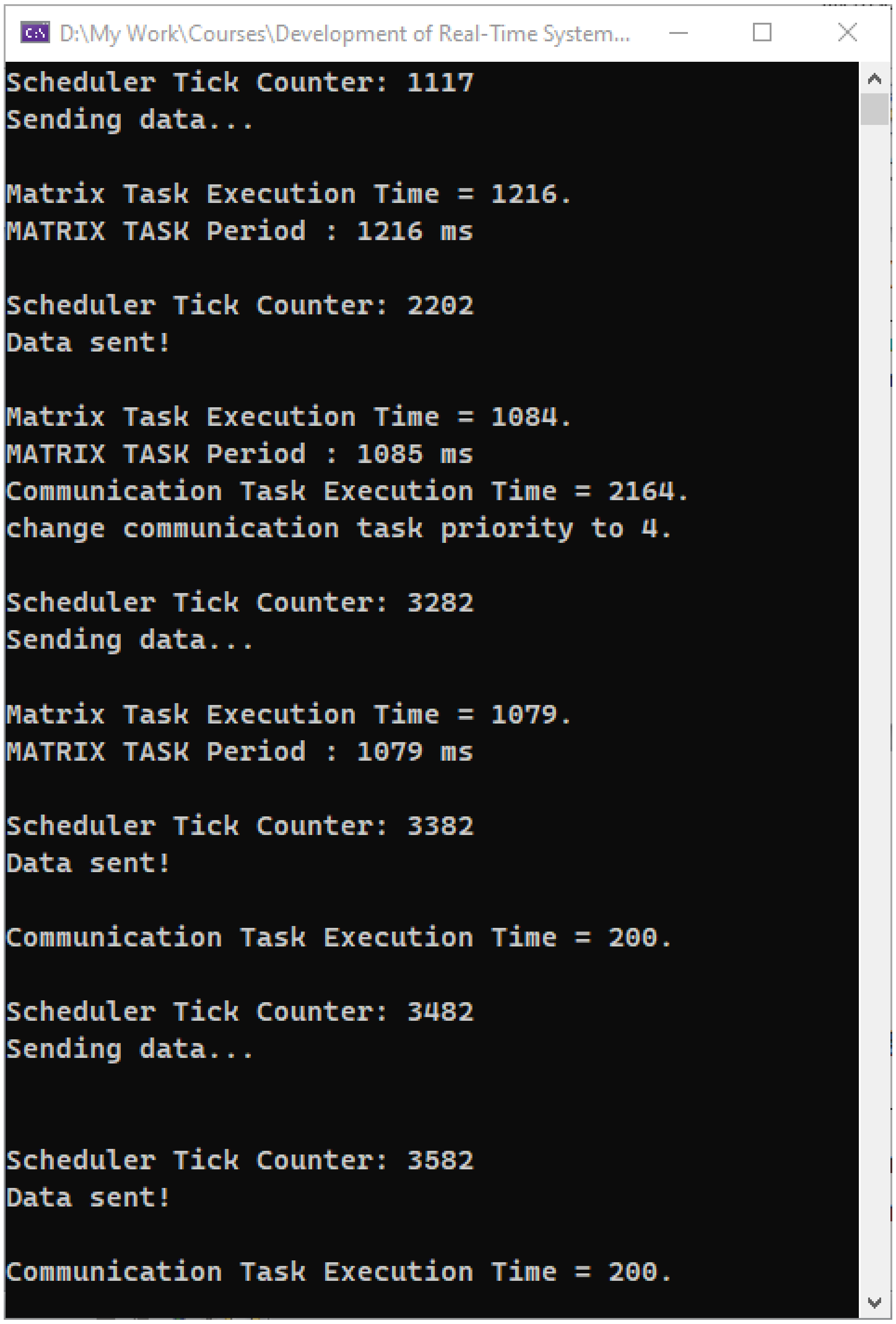
To measure matrix task period , I declare global variable to store the release time of matrix task Measured from count of ticks since vTaskStartScheduler was called and because here OS Tick is equal to one milli second , so number of Ticks equal to number of milli-seconds.

The period of "matrixtask" was about 1.1 seconds.

```
//release time of matrix task Measured from count of ticks since
vTaskStartScheduler was called.
volatile unsigned long matrix_release_time = 0;
```

in the beginning of the matrix task I calculate the period of task.

```
unsigned long current_tick_count = xTaskGetTickCount();
          printf("MATRIX TASK Period : %i ms \n",
abs(current_tick_count - matrix_release_time));
          fflush(stdout);
          matrix_release_time = current_tick_count;
```



```
Scheduler Tick Counter: 1117
Sending data...

Matrix Task Execution Time = 1216.
MATRIX TASK Period : 1216 ms

Scheduler Tick Counter: 2202
Data sent!

Matrix Task Execution Time = 1084.
MATRIX TASK Period : 1085 ms
Communication Task Execution Time = 2164.
change communication task priority to 4.

Scheduler Tick Counter: 3282
Sending data...

Matrix Task Execution Time = 1079.
MATRIX TASK Period : 1079 ms

Scheduler Tick Counter: 3382
Data sent!

Communication Task Execution Time = 200.

Scheduler Tick Counter: 3482
Sending data...


Scheduler Tick Counter: 3582
Data sent!

Communication Task Execution Time = 200.
```