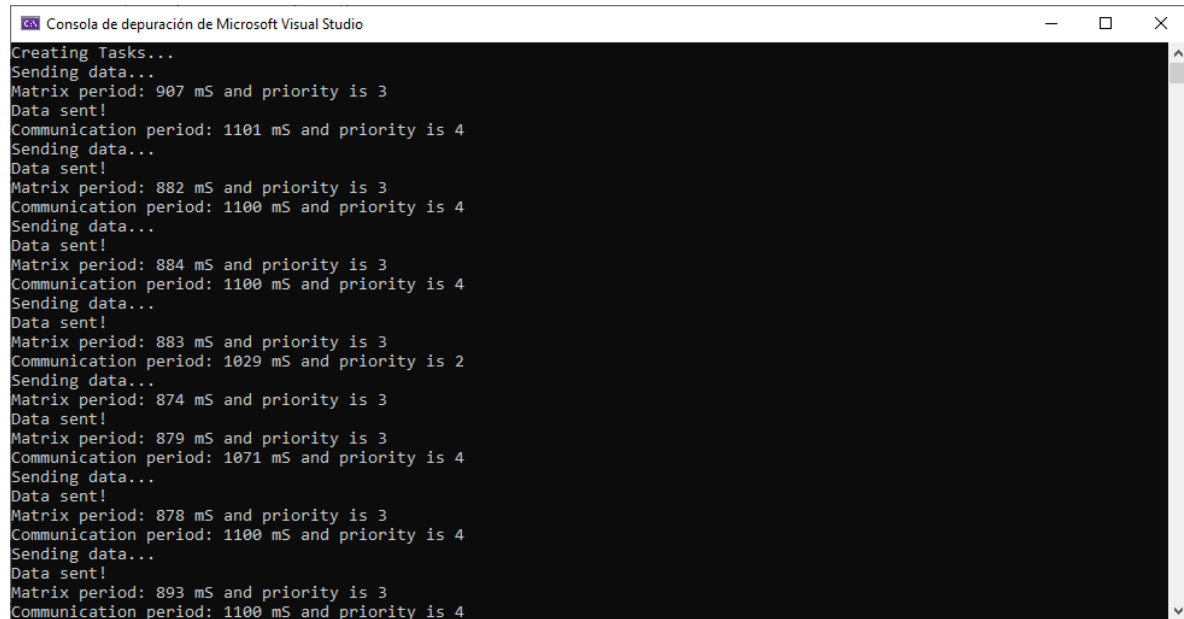


## Assignment 2 – assigning priorities

### Output

A screenshot of the Visual Studio debug console window. The title bar reads 'Consola de depuración de Microsoft Visual Studio'. The console output shows a sequence of messages: 'Creating Tasks...', 'Sending data...', 'Matrix period: 907 mS and priority is 3', 'Data sent!', 'Communication period: 1101 mS and priority is 4', 'Sending data...', 'Data sent!', 'Matrix period: 882 mS and priority is 3', 'Communication period: 1100 mS and priority is 4', 'Sending data...', 'Data sent!', 'Matrix period: 884 mS and priority is 3', 'Communication period: 1100 mS and priority is 4', 'Sending data...', 'Data sent!', 'Matrix period: 883 mS and priority is 3', 'Communication period: 1029 mS and priority is 2', 'Sending data...', 'Matrix period: 874 mS and priority is 3', 'Data sent!', 'Matrix period: 879 mS and priority is 3', 'Communication period: 1071 mS and priority is 4', 'Sending data...', 'Data sent!', 'Matrix period: 878 mS and priority is 3', 'Communication period: 1100 mS and priority is 4', 'Sending data...', 'Data sent!', 'Matrix period: 893 mS and priority is 3', 'Communication period: 1100 mS and priority is 4'.

```
Consola de depuración de Microsoft Visual Studio
Creating Tasks...
Sending data...
Matrix period: 907 mS and priority is 3
Data sent!
Communication period: 1101 mS and priority is 4
Sending data...
Data sent!
Matrix period: 882 mS and priority is 3
Communication period: 1100 mS and priority is 4
Sending data...
Data sent!
Matrix period: 884 mS and priority is 3
Communication period: 1100 mS and priority is 4
Sending data...
Data sent!
Matrix period: 883 mS and priority is 3
Communication period: 1029 mS and priority is 2
Sending data...
Matrix period: 874 mS and priority is 3
Data sent!
Matrix period: 879 mS and priority is 3
Communication period: 1071 mS and priority is 4
Sending data...
Data sent!
Matrix period: 878 mS and priority is 3
Communication period: 1100 mS and priority is 4
Sending data...
Data sent!
Matrix period: 893 mS and priority is 3
Communication period: 1100 mS and priority is 4
```

Matrix task has a period of about 900mS. Communication task has a period of about 1100mS, because matrix has all the priority until 1000mS have run and communication task takes the priority.

### Questions

- Why is "matrixtask" using most of the CPU utilization?  
Matrix has a higher priority, taking most CPU utilization.
- Why must the priority of "communicationtask" increase in order for it to work properly  
Communication task priority has to change to 4 to have a higher priority than matrix task (currently 3). This change to a higher priority allows execution of communication task.
- What happens to the completion time of "matrixtask" when the priority of "communicationtask" is increased?  
Completion time of matrix task increases
- How many seconds is the period of "matrixtask"? (Hint: look at `vApplicationTickHook()` to measure it)  
It takes about 900mS for the period of matrix task

## Code modifications

I defined these global variables

```
88 | xTaskHandle matrix_handle, communication_handle, priority_handle;
89 | int count = 0;
90 | int communication_count, matrix_count;
91 |
```

Matrix task has a modification to show the current period and to restart

```
140 |     printf("Matrix period: %d mS and priority is 3\n", matrix_count);
141 |     fflush(stdout);
142 |     matrix_count = 0;
```

Also communication task

```
156 |     int priority = uxTaskPriorityGet(communication_handle);
157 |     printf("Communication period: %d mS and priority is %d\n", communication_count, priority);
158 |     fflush(stdout);
159 |     communication_count = 0;
```

This is the prioritysettask()

```
void prioritysettask() {
    while (1) {
        if (communication_count > 1000)
            vTaskPrioritySet(communication_handle, 4);
        if (communication_count < 200)
            vTaskPrioritySet(communication_handle, 2);
        //printf("Communication count : %d \n", communication_count);
        vTaskDelay(100);
    }
}
```

Main code

```
int main( void )
{
    prvInitialiseHeap();

    vTraceEnable( TRC_START ); //esto faltaba

    printf("Creating Tasks...\n");

    // creating tasks for assignment
    xTaskCreate(matrix_task, "Matrix", 1000, NULL, 3, &matrix_handle);
    xTaskCreate(communication_task, "Communication", configMINIMAL_STACK_SIZE, NULL, 1, &communication_handle);
    xTaskCreate(prioritysettask, "priority_setter", configMINIMAL_STACK_SIZE, NULL, 5, &priority_handle);

    vTaskStartScheduler();
    for (;;)
        return 0;
}
```

And vApplicationTickHook()

```
273 void vApplicationTickHook( void )
274 {
275
276     //count++;
277     communication_count++;
278     matrix_count++;
279
280 }
281
282 /*
```