```c
/*
 * main.c
 *
 * Created: 24/09/2019
 * Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
 * Platform: Arduino Due / Atmel SAM3X8E
 * Purpose:  The complete program takes a string and prints it to the console (UART).
 *           The printing is carried out using two writer objects (writer1 and writer2),
 *           and  the  message is printed one symbol at a time. And the complete string is protected.
 *           The protected object (console_po) ensures mutual exclusion.
 *           Main.c is responsible for declaration, initializing board peripherals, calls
 *           all other init functions and starts the FreeRTOS scheduler.
 */

#include <asf.h>
#include <FreeRTOS.h>
#include <task.h>

#include <console_po.h>

#include <writer1.h>
#include <writer2.h>

int main (void)
{
    /* System clock initialization */
    sysclk_init();

    /* Board peripherals initialization */
    board_init();

    /* Init UART console */
    console_init();
    /* Init tasks */
    init_writer1();
    init_writer2();

    /* Start scheduler */
    vTaskStartScheduler();

    /* Should never reach here ... */
    return 1;
}
```

```c
/*
* console_po.c
*
* Created: 24/09/2019
* Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
* Platform: Arduino Due / Atmel SAM3X8E
* Protected object - UART0 module
*
* For configuration of the UART see conf_uart_serial.h!
* Ensures mutual exclusion when printing to the UART
*/

/* Imported libraries */
#include <asf.h>
#include <FreeRTOS.h>
#include <task.h>

#include <console_po.h>

/* Required interfaces */
#include <stdlib.h>
#include <stdio.h>

/* Semaphore declaration */
SemaphoreHandle_t xSemaphore;


void printfConsole(const char * cStr) {

    xSemaphoreTake( xSemaphore, (TickType_t) 1) == pdTRUE;
        /* If we're able to access the semaphore then the task gains access to
          the shared resources */

    for(int i = 0; (cStr[i] != '\0'); i++) {
        /* For loop to iterate through the string in writer character by
          character */


            /* Checks if the microcontroller is ready to send the next message
              */
            addChar(cStr[i]);


    }
    xSemaphoreGive(xSemaphore);     /* Free up the semaphore for other tasks */


}

void addChar(unsigned char c) { /* If the microcontroller is ready it prints the
   element of the string */
    CONF_UART->UART_THR = c;
}
```

```
void console_init()
{
    xSemaphore = xSemaphoreCreateMutex(); /* Creates a mutual exclusion
      semaphore for use by the tasks */

    const usart_serial_options_t usart_serial_options = {
        .baudrate   = CONF_UART_BAUDRATE,
        .charlength = CONF_UART_CHAR_LENGTH,
        .paritytype = CONF_UART_PARITY,
        .stopbits   = CONF_UART_STOP_BITS
    };


    stdio_serial_init(CONF_UART, &usart_serial_options);

}
```

```c
/*
* console_po.h
*
* Created: 24/09/2019
* Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
* Platform: Arduino Due / Atmel SAM3X8E
* Protected object - UART0 module
*
* For configuration of the UART see conf_uart_serial.h!
* Ensures mutual exclusion when printing to the UART
*/

#ifndef UART_COMM_H_
#define UART_COMM_H_

/**
 *  Configure and initialize the Console UART.
 */
void console_init(void);

/**
 * Print function, ensures mutual exclusion
 */
void printfConsole(const char *);

void addChar(unsigned char);
#endif /* UART_COMM_H_ */
```

```c
/*
 * writer1.c
 *
 * Created: 24/09/2019
 * Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
 * Platform: Arduino Due / Atmel SAM3X8E
 * Purpose:  Initialises the vTaskwriter1 task which sends message1 to the console
 */


#include <asf.h>
#include <FreeRTOS.h>
#include <task.h>
#include <console_po.h>
#include <writer1.h>

/* Declares the object writer1 for later definition of tasks */
static void writer1(void*); /* Declared static to protect the function from being accessed by other objects*/

static void writer1 (void *pvParameters) {
    portTickType xLastWakeTime ;
    xLastWakeTime = xTaskGetTickCount();
    char cStr[] = "Vad ";   /* Creates the string to be printed to the console */

    for(;;) {

        printfConsole(cStr);    /* Calls the printing function defined in the console_po.c file */
        vTaskDelayUntil(&xLastWakeTime, (WRITER1PERIOD/portTICK_PERIOD_MS)); /* Tells the controller to wait before switching to the next task to prevent overlapping of tasks */
    }

}

void init_writer1() {
    /* Creates the writer1 task */
    xTaskCreate(
    writer1,                    /* Function that implements the task. */
    "Message 1 print task",     /* Text name for the task. */
    250,                        /* Stack size in words, not bytes. */

    NULL,                       /* Parameter passed into the task. */
    1,                          /* Priority at which the task is created. */
    NULL                        /* Used to pass out the created task's handle. */
    );
}
```

```
/*
* writer1.h
*
* Created: 24/09/2019
* Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
* Platform: Arduino Due / Atmel SAM3X8E
* Purpose:  Initialises the vTaskwriter1 task which sends message1 to the
  console
*/


#ifndef WRITER1_H
#define WRITER1_H

#define WRITER1PERIOD 1

/* Initializes the writer1 task for use by the console function */
void init_writer1(void);

#endif /* writer1_H */
```

```c
/*
 * writer2.c
 *
 * Created: 24/09/2019
 * Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
 * Platform: Arduino Due / Atmel SAM3X8E
 * Purpose:  Initialises the vTaskwriter2 task which sends message2 to the console
 */

#include <asf.h>
#include <FreeRTOS.h>
#include <task.h>
#include <console_po.h>
#include <writer2.h>

/* Declares the object writer2 for later definition of tasks */
static void writer2 (void*);


static void writer2 (void *pvParameters) {/* Declared static to protect the function from being accessed by other objects*/

    portTickType xLastWakeTime ;
    xLastWakeTime = xTaskGetTickCount();
    char cStr[] = "Bra! \n";    /* Creates the string to be printed to the console */

    for(;;) {

        printfConsole(cStr);    /* Calls the printing function defined in the console_po.c file */
        vTaskDelayUntil(&xLastWakeTime, (WRITER2PERIOD/portTICK_PERIOD_MS)); /* Tells the controller to wait before switching to the next task to prevent overlapping of tasks */
    }

}



void init_writer2() {
    /* Creates the writer2 task */
    xTaskCreate(
    writer2,                    /* Function that implements the task. */
    "Message 2 print task",     /* Text name for the task. */
    250,                        /* Stack size in words, not bytes. */
    NULL,                       /* Parameter passed into the task. */
    1,                          /* Priority at which the task is created. */
    NULL                        /* Used to pass out the created task's handle. */
    );
}
```

```
/*
 * writer2.h
 *
 * Created: 24/09/2019
 * Author:  Cornelis Peter Hiemstra, Noel Janes & Flavia Pérez Cámara
 * Platform: Arduino Due / Atmel SAM3X8E
 * Purpose:  Initialises the vTaskwriter2 task which sends message2 to the
   console

 */

#ifndef WRITER2_H
#define WRITER2_H

#define WRITER2PERIOD 1

/* Initializes the writer2 task for use by the console function */
void init_writer2(void);

#endif /* writer2_H */
```