Lab 5 Report
CSE 4560 Embedded System

Name: Daniel Delgado Acosta Coyote ID:006896598

I Goals

Goals of this Lab

- **Introduce a real-time operating system**
- **Learn what tasks are and how to create them**
- **A look at task states and priorities**
- **Inter-task communication using queues**
- **Critical sections**

II Lab Questions, Processes and Program

Task 1: Task Creation

Task 1.1: The first task

What can you see?

The terminal prints task A and B statuses, however, it is out of order.

Task 1.2: Creating a second task

What can you observe? Can you explain why it might not work properly?

After this task, both task A and B output correct information. Before using one function to output both task, perhaps what went wrong was that the program was printing each task in the order of execution.

Task 2: Critical Sections and Priorities

Task 2.1: Mutex

In which order are the tasks executed?

After completing this task, task A was executed first then task B because task A was set as priority.

Task 2.2: Priorities

What can you observe?

Now, Task B is the only task being executed.

Task 2.3: Design considerations for the real-time operating system

Considering the predictability, reliability, security, efficiency, explain why the performance of this real-time operating system is critial to the insulin pump controler in terms of public health, safety and welfare. Please also discuss the impacts of global, cultural, social, environmental, and economic factors on the design of such an embedded system for insulin pump.

Insulin is a horomone that needs to be balanced in the body therefore pumping over a set amount of time is crucial to the patient, therfore the real-time operating system must not fail.

Depending on the overall system, an embedded system for an insulin pump task should be easy to integrate and low cost.

III Answers and Results

Task 1.1: The first task

Changed Code

```
77 void vTaskAFunction(void *pvParameters) {
78     char *message = (char*)pvParameters;
79     uart_println("Start TaskA.");
80 
81     while (1) {
82         uart_println(message);
83         vSimpleDelay();
84     }
85 }
86 
87 // TODO: Define your functions here
88 uart_println ("Creating TaskA.");
89 char * pcParameters1 = "TaskA is running and running.";
90 result = xTaskCreate (vTaskAFunction, " TaskA ", 1000, (void*)pcParameters1, 1, NULL);
91 if (result != pdPASS)
92 {
93 uart_println ("Error creating TaskA task.") ;
94 return 0;
95 }
```

Explanation: Prints Task A status to serial monitor.

Task 1.2: Creating a second task

Changed Code

```
52    uart_println("Creating TaskB.");
53    char *pcParameters2 = "TaskB is running and running.";
54    result =  xTaskCreate(vTaskBFunction, "TaskB", 1000, (void*)pcParameters2, 1, NULL);
55    if (result != pdPASS) {
56        uart_println("Error creating TaskB task.");
57        return 0;
58
59
60    // TODO: Insert your code here
61        xTaskCreate (vTaskAFunction , "TaskB", 1000, (void*)"TaskB is running and running.", 1, NULL);
```

Explanation: Task B will display if its running.

Task 1.3: Handles and deleting a task

Changed code

```
// TODO: Declare your functions here
int counter = 0 //counter
TaskHandle_t xTaskAHandle = NULL; //Handle used to delete task A
```

```
int main(void) {
    // Initialize UART
    uart_init(uart_baudrate);

    // TODO: Insert your code here
    void vTaskBFunction (void * pvParameters)
    {
    char * message = (char*) pvParameters;
    uart_println ("Start TaskB.");

    while (1) {
    uart_println ("%d:", counter);
    uart_println (message);
    if( counter == 10) {
    vTaskDelete (xTaskAHandle);
    }
    else if(counter == 20) {
    vTaskDelete (NULL);
    }
    counter ++;
    vSimpleDelay ();
    }
    }
```

Explanation: Task can be deleted using new function.

Task 2.1: Mutex

Changed code

```
21 SemaphoreHandle_t xMutex;
22 xMutex = xSemaphoreCreateMutex();
```

```
135 void uart_println_mutex(const char* str, ...)
136 {
137
138     xSemaphoreTake ( xMutex , portMAX_DELAY );
```

Explanation: Sets task A as priority.

IV Problem

The main and only issue I had with this lab was that the file itself seemed to have many missing or incorrect code. I had to fix a lot of it by going of my errors when debugging.

V What have you learned

During this lab, I learned about setting up tasks as void function and calling other void function to either delete a task or prioritize one over another.