

Overview

- Create two tasks (Task1 , Task2)
- Print “Task1” every 100 ms
- Print “Task2” every 500 ms
- Task1 have priority of 3 and stack size of 1000
- Task2 have priority of 1 and stack size of 100

The image consists of two screenshots of the Visual Studio Code editor, showing the development and execution of a FreeRTOS application.

Top Screenshot: The editor displays the `main.c` file. The code defines two tasks, `Task1` and `Task2`, and a `main` function. `Task1` is a simple loop that prints "This is task 1" every 100ms. `Task2` is a simple loop that prints "This is task 2" every 500ms. The `main` function initializes the FreeRTOS scheduler and starts the tasks.

```
src > C main.c > task1(void)
1 #include "FreeRTOS.h"
2 #include "task.h"
3
4
5 void Task1(void)
6 {
7     while (1)
8     {
9         printf("This is task 1 \n");
10        vTaskDelay(100);
11    }
12 }
13
14
15
```

The terminal shows the output of the program, which is a sequence of "This is task 1" and "This is task 2" messages, indicating that the tasks are running in parallel.

```
PS E:\RTOS\tasks-win32-mvc\build> .\main.exe
This is task 1
This is task 2
This is task 1
This is task 1
This is task 1
This is task 1
This is task 1
This is task 1
This is task 2
This is task 1
```

Bottom Screenshot: The editor displays the `main.c` file. The code defines two tasks, `Task1` and `Task2`, and a `main` function. `Task1` is a simple loop that prints "This is task 1" every 100ms. `Task2` is a simple loop that prints "This is task 2" every 500ms. The `main` function initializes the FreeRTOS scheduler and starts the tasks.

```
src > C main.c > main(void)
1 #include "FreeRTOS.h"
2 #include "task.h"
3
4
5 void Task1(void)
6 {
7     while (1)
8     {
9         printf("This is task 1 \n");
10        vTaskDelay(100);
11    }
12 }
13
14
15
16
17 void Task2(void)
18 {
19     while (1)
20     {
21         printf("This is task 2 \n");
22        vTaskDelay(500);
23    }
24 }
25
26
27 int main(void)
28 {
29     xTaskHandle HT;
30     xTaskCreate(Task1, "HelloTask", 1000, NULL, 3, &HT);
31     xTaskCreate(Task2, "HelloTask", 100, NULL, 1, &HT);
32     vTaskStartScheduler();
33     for(;;);
34     return 0;
35 }
36
```