

Exercícios de Fixação

Filas são usadas em sistemas embarcados para sincronizar a execução de TASKS e para passar variáveis entre TASKS, reduzindo a necessidade de variáveis globais e portanto condições de corrida. As operações em filas no FreeRTOS são atômicas, ou seja, quando uma TASK escreve na fila ou lê da fila, nenhuma outra thread interrompe. Por exemplo, na Figura 1, a TaskA escreve na fila. A TaskC depois pode escrever, e assim, por diante. Cada novo dado é inserido “atrás” do anterior, no conceito FIFO (first in, first out).

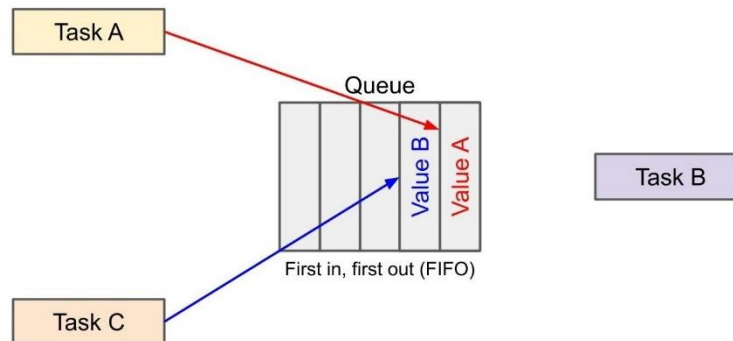
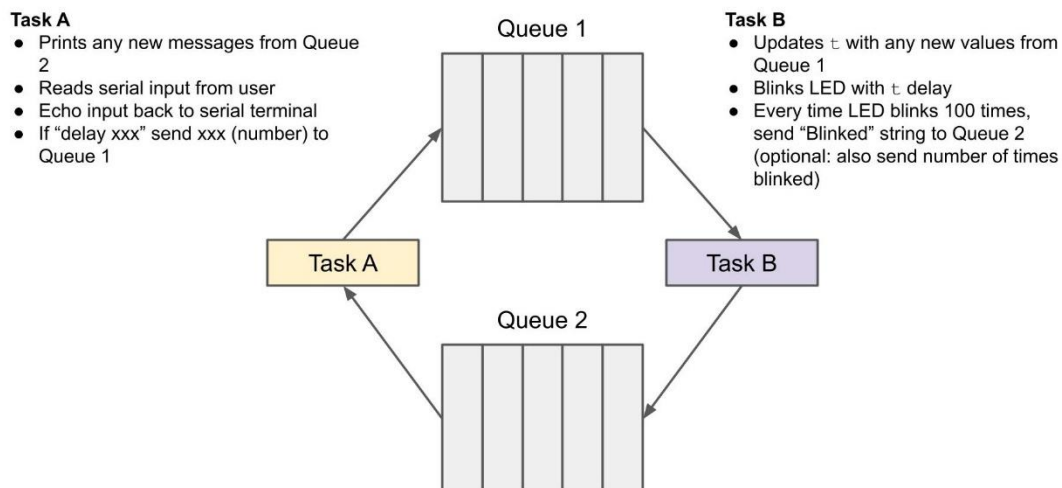


Figura 1

Lembrando que os valores são passados para a fila por valor e não por referência. Então todo o conteúdo/variável é copiada atômicamente.

Tarefa: tomando por base os exemplos vistos em sala de aula sobre filas disponíveis em https://github.com/josenalde/parallel_programming_rtos/tree/main/src, leitura de caracteres/valores pelo Serial Monitor e os conceitos de produtor-consumidor, considere o seguinte problema, com 02 Tasks e 02 Filas:



Requisitos funcionais:

1) TaskA:

- a. imprime qualquer nova mensagem que está disponível na Fila 2
- b. lê entradas da serial fornecidas pelo usuário e as exibe na própria serial (Serial Monitor). Se na entrada tiver a palavra **delay** seguida por **espaço** seguida por um **número**, este número deve ser enviado para a Fila 1. Sugestão de usar vetor de char e Serial.read para ler caracter a caracter e ir adicionando ao vetor (buffer) ou usar Serial.readString() e colocar o conteúdo em variável String:
<https://www.delftstack.com/pt/howto/arduino/arduino-serial-read-string/>

2) TaskB:

- a. lê mensagens da Fila 1. Se for um número (ms), atualiza o tempo de delay da piscada do LED com este número
- b. pisca o LED (pode ser o interno) com este delay
- c. a cada 10 piscadas do LED, a mensagem contendo a palavra “pisquei o LED x vezes” deve ser gravada na Fila 1. Considere usar o sprintf para montar mensagens, tal como nos exemplos vistos no slide09.

Preparar este código para teste prático com o ESP32 nos dias 16 e 17.

As duas tarefas podem ser associadas ao mesmo núcleo 1, com prioridade 1.

Dentro do setup(), a única linha deve ser vTaskDelete(NULL), para que as tarefas setup() e loop() não sejam mais executadas.

OBS: existem simuladores do FreeRTOS para Windows: <https://www.freertos.org/FreeRTOS-Windows-Simulator-Emulator-for-Visual-Studio-and-Eclipse-MingW.html>, mas o código precisa de ajustes, pois há diferenças no port do FreeRTOS para ESP32 da espressif, bem como para Linux: <https://www.freertos.org/FreeRTOS-simulator-for-Linux.html>.

Quem tiver acesso a placa ESP32 ou mesmo à placa Arduino Uno, pode testar. Neste último caso, instalar o FreeRTOS para Arduino UNO importando a biblioteca na IDE do Arduino.

Dúvidas enviem e-mail para josenalde.oliveira@ufrn.br