



TÓPICOS ESPECIAIS – SISTEMAS EMBARCADOS

PROGRAMAÇÃO PARALELA E TEMPO REAL

PROF. JOSENALDE OLIVEIRA

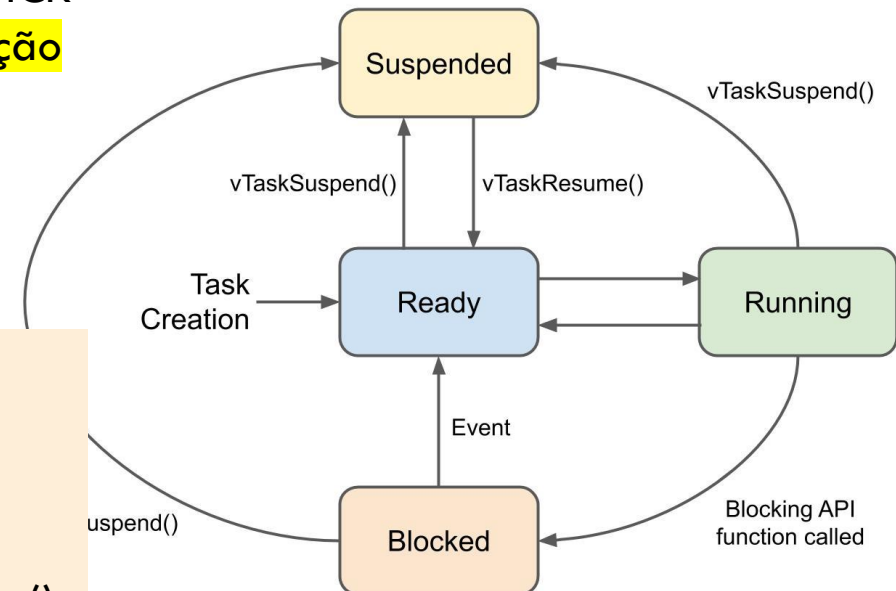
josenalde.oliveira@ufrn.br

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN

FreeRTOS – lidando com TASKS

- No FreeRTOS, a fatia de tempo (tick) é 1 ms
- A cada 1 ms, o temporizador de hardware cria uma interrupção. A ISR do temporizador roda o scheduler, escolhe a próxima tarefa a executar
- A cada interrupção (tick), a tarefa com maior prioridade é escolhida
- Se as tarefas de maior prioridade possuem a mesma prioridade, são executadas em round-robin (divisão de tempo)
- Se uma task de maior prioridade que a task e em execução entra em modo Task States READY, é executada imediatamente, sem esperar próximo TICK
- Uma ISR tem prioridade maior que qualquer task em execução

Criação – Pronta (Ready) – Scheduler escolhe e coloca em running
Pode retornar ao Ready, ou ser bloqueada por um `vTaskDelay` ou
Por que está aguardando algum recurso, como um semáforo. Se
bloqueada outras tasks podem rodar. Qualquer thread pode colocar
a si e outras em suspenso (`vTaskSuspend()`) e retornar com `vTaskResume()`



FreeRTOS – lidando com TASKS

- Lendo serial e piscando LEDs

https://github.com/josenalde/parallel_programming_rtos/blob/main/src/serial_led_tasks/serial_led_tasks.ino

- alterar prioridade do Led para 2, mantendo a Serial em 1, o que ocorre?
- alterar prioridade do Serial para 2, mantendo Led em 1, o que ocorre?

FreeRTOS – lidando com TASKS

- Variável local perdendo escopo antes de ser passada à task – proteger com Mutex/Semáforo
https://github.com/josenalde/parallel_programming_rtos/tree/main/src/led_serial_mutex_1

- Usando mutex para proteger a criação da task e passagem do parâmetro

https://github.com/josenalde/parallel_programming_rtos/blob/main/src/led_serial_mutex_2/led_serial_mutex_2.ino

- Alternando LEDs com 2 tasks e mutex

https://github.com/josenalde/parallel_programming_rtos/blob/main/src/alternate_leds/alternate_leds.ino