Universidade Federal de Juiz de Fora - Faculdade de Engenharia

Atividade 1: Tasks

Professor: Leandro Manso **Disciplina:** CEL080 - Software Embarcado

Exercício 1: Faça um software que contenha 4 tasks de mesma prioridade. Cada tasks deve ter as seguintes características:

- Devem poder executar tanto na CPU0 como na CPU1;
- Devem realizar uma chamada de bloqueio para um delay de 100ms.
- Devem imprimir, utilizando a biblioteca esp_log quando estiverem executando e em qual núcleo estão executando. Utilize a função: BaseType_t xPortGetCoreID(void)

Exercício 2: No software do exercício anterior, adicione uma carga de processamento em cada task como mostrado no exemplo abaixo.

```
for (ii = 0; ii < lim; ii++)
{
  for (jj = 0; jj < 50000; jj++)
  {

  }
}
```

- Estabeleça o valor da variável lim para que o escalonador comece a utilizar os dois núcleos de processamento.
- Altere a criação das tasks de forma que as tasks 1 e 2 tenham afinidade com a CPU0 e as tasks 3 e 4 tenham afinidade com a CPU1.

Exercício 3: No software do exercício anterior verifique como está a utilização da stack de cada task. Utilize a função UBaseType_t uxTaskGetStackHighWaterMark(TaskHandle_t xTask)

Exercício 4: No software do exercício anterior faça com que a após 10 rodadas no loop infinito da task 1, ela delete a task2 e após 20 rodadas no loop infinito da task 3, ela delete a task 4.

- Para isso, utilize a função void vTaskDelete(TaskHandle_t xTaskToDelete)
- O parâmetro de entrada da função é uma variável do tipo TaskHandle_t que deve ser criada e passada como o sexto argumento da função xTaskCreate ou xTaskCreatePinnedToCore na criação das tasks.