Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Eletrônica (DAELN)

SISTEMAS EMBARCADOS

PROJETO

Prof. André Schneider de Oliveira

andreoliveira@utfpr.edu.br

Objetivo

- Escrever uma programa em linguagem C++ Cortex-M, utilizado o CMSIS RTOS, para o gerenciamento de múltiplas tarefas matemáticas com prioridades dinâmicas

Tarefa	Funcionalidade	Prioridade	Deadline	Periodicidade
A	$\sum_{x=0}^{256}[x+(x+2)]$	low (10)	70%	8Hz
B	$\sum_{n=1}^{16}(2^n/n!)$	normal (0)	50%	2Hz
C	$\sum_{n=1}^{72}[(n+1)/n]$	high (-30)	30%	5Hz
D	$1+(5/3!)+(5/5!)+(5/7!)+(5/9!)$	normal (0)	50%	1Hz
E	$\sum_{x=1}^{100}(x\pi^2)$	high (-30)	30%	6Hz
F	$\sum_{y=1}^{128}(y^3/2^y)$	**realtime (-100)**	10%	10Hz

Especificações

- O sistema deve executar periodicamente as operações matemáticas especificadas para cada tarefa, de acordo com sua frequência

- Deve-se atribuir um sistema de prioridades onde a de **menor valor é a mais alta**

- É necessária a inclusão de uma fila de execução que armazene o identificador, prioridade e tempo de relaxamento restante (em ticks) das threads

Especificações

- Toda a tarefa possui o seu deadline especificado, ou seja, o percentual a mais de **ticks** que pode aguardar até a tarefa ser concluída, por exemplo

tarefa leva 20 ticks e possui um deadline 30%

a tarefa deve terminar em até 26 ticks após ser iniciada

***Sugestão: executar a operação em um programa simples (sem RTOS) para determinar o tempo de execução em ticks**

Especificações

- Tarefas com prioridade **realtime** não podem finalizar após o seu deadline. Caso ocorra, o sistema terá falha geral e irá encerrar com a mensagem de erro ***"master fault"***

- Caso alguma tarefa com prioridade **não-realtime** seja finalizada **após o seu deadline**, a mesma deve ter a sua **prioridade incrementada** na próxima execução, gerando a mensagem **"secondary fault"**

- Caso alguma tarefa com prioridade **não realtime** seja finalizada **antes da metade do seu deadline**, a mesma deve ter sua **prioridade decrecida** na próxima execução, gerando a mensagem **"secondary fault"**

Especificações

- As informações de execução das tarefas e do escalonador devem ser apresentadas no Debugger, dentre elas

  - **Prioridade**
  - **Tempo de relaxamento restante (em ticks)**
  - **Estado *(ready, running, waiting)***
  - **Percentual de execução (quanto já foi processada)**
  - **Atraso (em ticks)**
  - **Fila de execução**
  - **Faltas ocorridas**

Especificações RTOS

- A implementação deve se basear no CMSIS RTOS, onde cada tarefa deve ser necessariamente uma thread
- A thread principal (main) deve ser convertida em escalonador após a inicialização do ***Kernel***
- Deve ser respeitada a nomenclatura das threads
- **Sugestão:** a utilização de um semáforo no escalonador pode minimizar a ocorrência de execuções não desejadas das tarefas