



### Dados do Aluno

Matrícula	Turma	Aluno
22101934	06605	Kamylo Serafim Porto

### PROJETO 3 - Escalonador de tarefas aperiódicas e acesso a recursos compartilhados

Implementações: RM, Background Scheduling e NPP.

#### 1) Implementação das tarefas aperiódicas

As tarefas aperiódicas quando inicializadas são alocadas em fila no array *OS\_aperiodic\_tasks*. Quando elas são finalizadas, a função *OS\_finished\_aperiodic\_task()* é chamada para reorganizar essa fila.

De maneira conceitual, quando **nenhuma** tarefa periódica estiver pronta para ser executada, o escalonador verifica a existência de tarefas aperiódicas a serem escalonadas. Caso isso seja verdadeiro, as aperiódicas são escalonadas por ordem de chegada (first-in-first-out). Caso contrário, a thread Idle é escalonada.

#### 2) Implementação do gerenciamento de recursos

O gerenciamento e controle de acesso aos recursos e definições de prioridades são realizados nas funções *sem\_down()* e *sem\_up()*.

Cada tarefa possui um array de prioridades para o controle de acesso a regiões aninhadas. Quando uma região crítica é acessada, as prioridades são deslocadas em 1 posição para que a posição 0 receba a prioridade da região crítica acessada. Como a implementação é NPP, essa prioridade é a maior do sistema.

Quando uma tarefa (periódica ou não) entra em uma região crítica o seu ponteiro (*OSThread \**) é adicionado na posição de maior prioridade do array *OS\_tasks* e o bit correspondente no bitmask de tarefas prontas para serem

escalonadas (*OS\_readySet*) é definido como 1.

### **3) Teste da implementação**

Para o teste da implementação foram propostas 2 tarefas periódicas e 2 aperiódicas que compartilham o acesso à regiões críticas A e B. Os detalhes, comentários e funcionamento sobre a implementação dos testes são encontrados em um vídeo neste link: <https://www.youtube.com/watch?v=uwlmY5ckUzg>