



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

T569 –SISTEMAS DE TEMPO REAL

Aula 10

Prof. Marcelo Sousa



Agenda

- Escalonadores Clock Driven
 - Table-Driven
 - Cíclico
 - Escalonador generalizado
- Escalonadores híbridos
 - Time Sliced Round Robin
- Escalonadores Event Driven
 - Foreground-Background



Escalonador Clock Driven

- Baseia sua decisão a respeito de qual *task* será executada apenas em pontos de interrupção baseados em clock.
- Os pontos de escalonamento são determinados pelas interrupções do timer



Escalonador Clock Driven

- Também conhecidos com escalonadores **Off-Line**, pois antes de iniciar o sistema propriamente dito o escalonamento de tarefas é fixado, ou seja, a ordem de execução das tarefas é definida previamente.



Escalonador Clock Driven

- Vantagem:
 - Pouco *overhead* para tratamento de agendamento de tarefas
- Desvantagem:
 - Pouca flexibilidade para o escalonamento de tarefas aperiódicas e esporádicas, uma vez que seu acontecimento não é preditivo.



Escalonador Table Driven

- Agenda o tempo em que cada tarefa será executada em uma tabela.

Task	Start Time in milli Seconds
T_1	0
T_2	3
T_3	10
T_4	12
T_5	17



Escalonador Table Driven

- Qual o tempo de repetição da tabela do escalonador?
 - $ST = \{T_i\} \rightarrow$ Conjunto de n tarefas
 - $p_i \rightarrow$ é o período da tarefa T_i
 - $MMC(p_1, p_2, p_3, \dots, p_n) \rightarrow$ Tempo de repetição do escalonador (*major cycle*)



Escalonador Table Driven

- Exemplo:
 - $T1 = (e1 = 5\text{ms}, p1 = 20\text{ms})$
 - $T2 = (e2 = 20\text{ms}, p2 = 100\text{ms})$
 - $T3 = (e3 = 30\text{ms}, p3 = 250\text{ms})$
- Qual o tempo de repetição do escalonador?
 - $\text{MMC}(20, 100, 250) = 500\text{ms}$



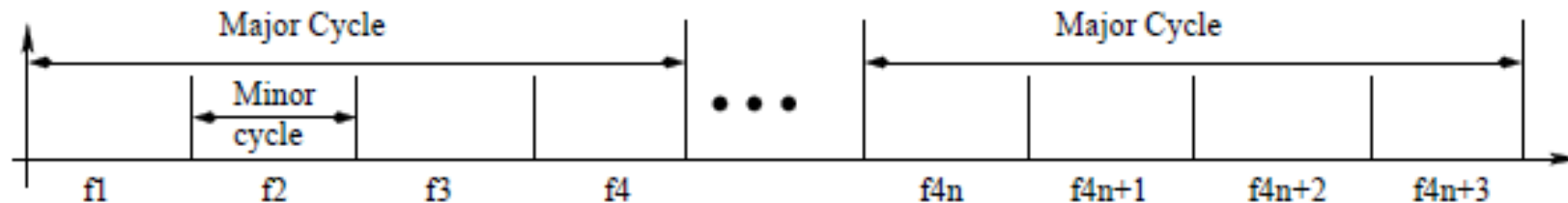
Escalonador Cíclico

- São escalonadores bastante populares e utilizados largamente na indústria.
- São escalonadores de fácil programação
- Também utilizam uma tabela pré-definida.
- O tempo de repetição também é dado pelo *major cycle*



Escalonador Cíclico

- O *major cycle* é dividido em *minor cycles*
- Cada *minor cycle* é chamado de *frame*





Escalonador Cíclico

- Definição do tempo do Frame
 - 1) Tempo mínimo de troca de contexto
 - $\max\{e_i\} \leq F$
 - 2) Tamanho mínimo da Tabela
 - O macro ciclo deve ser múltiplo do micro ciclo
 - 3) Atendimento aos deadlines
 - $2F - \text{MDC}(F, p_i) \leq d_i$
 - Matematicamente, garante que haverá um *frame* completo entre o momento de chegada de uma task e o deadline



Escalonador Cíclico

- Exemplo: Calcular o tamanho do *frame size* para o conjunto de tarefas abaixo
 - $T1=(e1=1, p1=4)$
 - $T2=(e2=1, p2=5)$
 - $T3=(e3=1, p3=20)$
 - $T4=(e4=2, p4=20)$



Escalonador Cíclico

- Atendimento às restrições:
 - 1) $\max\{e_i\} \leq F \rightarrow F \geq 2$
 - 2) Macro ciclo $\rightarrow \text{MMC}(4,5,20) = 20$
 - Valores possíveis $\rightarrow 2, 4, 5, 10, 20$
 - 3) Testar todas os valores para a inequação:
 - $2F - \text{MDC}(F, p_i) \leq d_i$
 - Valor adequado: 2



Escalonadores generalizados

- Atendimento de tarefas esporádicas e aperiódicas por escalonadores cíclicos.
- Utilização do tempo livre de cada frame para tratamento de eventos esporádicos ou aperiódicos
- *Frames* possuirão tempo ocioso sempre que o tempo de execução de uma tarefa for menor que o tempo do frame.



Escalonadores generalizados

- Sempre que houver a chegada de uma tarefa esporádica, é realizado um teste de aceitação para avaliar se é possível atender aquela tarefa com o tempo disponível por aquele frame.



Comparação entre métodos

- Inicialização do Timer
 - Cyclic: Apenas uma vez na inicialização do sistema
 - Table: A cada inicio de tarefa é necessária uma nova configuração:
- O timer Cyclic é mais eficiente, talvez por isso seja mais utilizado.



Escalonadores Híbridos

- Basicamente, são escalonadores que possuem seus pontos de escalonamento baseados em temporizadores ou eventos.



Escalonadores Híbridos

- Time Sliced Round Robin Scheduling
 - Características:
 - Escalonamento Preemptivo
 - As tarefas em estado “*ready*” são mantidas em uma fila circular, onde as tasks são executadas uma após a outra, obedecendo a ordem da fila
 - Uma vez a tarefa capturada, ela é executada por um período de tempo fixo e alocada novamente na fila circular.
 - Trata todas as tarefas com a mesma prioridade, mas uma pequena variação é possível. Por exemplo, dedicar mais *time slices* para tarefas mais prioritárias.



Escalonadores Event Driven

- Os pontos de escalonamento são definidos por eventos, tais como, finalização de uma tarefa, chegada de uma interrupção, ...
- São mais proficientes no tratamento de tarefas esporádicas e aperiódicas.
- Em contra partida, seus algoritmos são mais complexos.
- Este tipo de escalonamento normalmente é preemptivo.



Escalonadores Foreground-Background

- Classificação das tasks
 - Foreground Task: Tarefas de tempo real
 - Background Task: Tarefas não tempo real, esporádicas e aperiódicas
- Funcionamento:
 - Entre as **foreground tasks**, a cada ponto de escalonamento a tarefa de maior prioridade é escalonada para execução.
 - As **background tasks** só serão executadas caso não haja nenhuma foreground task no estado “ready”



Escalonadores Foreground-Background

- Em um sistema de tempo real, existem n foreground tasks periódicas definidas como: T_1 , T_2 , ..., T_n . Apenas a task T_B é de background e e_B o tempo de processamento de T_B .
- Onde o tempo em que a tarefa será finalizada (*completion time*) é dado por:

$$c_{t_B} = \frac{e_B}{1 - \sum_{i=1}^n \frac{e_i}{p_i}}$$



Escalonadores Foreground-Background

- Exemplo: Considere o seguinte sistema e calcule o *completion time* da tarefa T_b :

$$T_f = (\phi_f = 0, e_f = 50ms, p_f = 100ms, d_f = 100ms)$$

$$T_B = (e_B = 1000ms)$$

- Resp: 2000ms



Escalonadores Foreground-Background

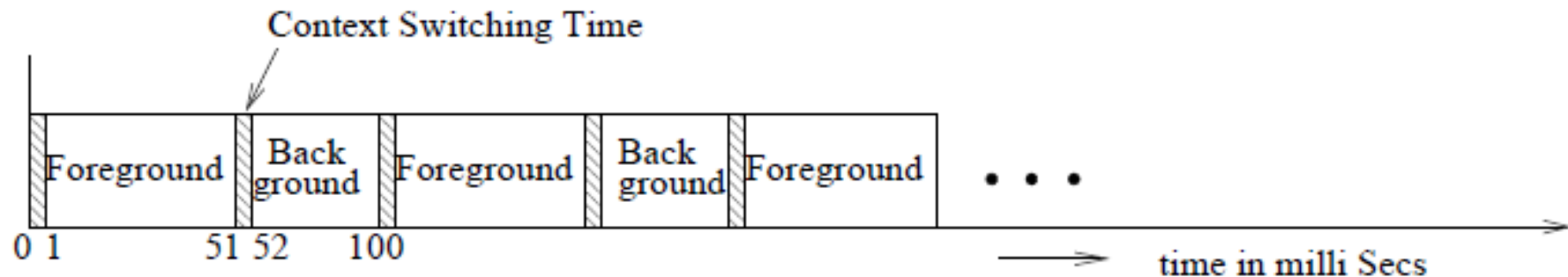
- Exemplo: Em um sistema com escalonamento baseado em prioridades e preemptivo duas foreground tasks T1 e T2 e uma background task são escalonadas. A task T1 periódica possui maior prioridade é executada a cada 20 ms e leva 10ms para completar sua tarefa. T2 necessita de 20ms a cada 50ms. T3 é uma background task e necessita de 100ms para ser completa. Considerando que todas as tarefas estão prontas para serem executadas em quanto tempo a tarefa T3 terminará sua execução?

- Resp: 1000ms



Escalonadores Foreground-Background

- Exemplo: Considere que é posto um *overhead* de 1ms a cada troca de contexto de tipos de tarefas, calcule o *completion time* de Tb.



- Resp: 2083,4ms



Próxima Aula

- Escalonadores:
 - Earliest Deadline First (EDF)
 - Rate Monotonic Algorithm