Universidade Tecnológica Federal do Paraná – Toledo
Engenharia da Computação – COENC

Sistemas Embarcados

Escalonamento de tarefas

Tiago Piovesan Vendruscolo

Escalonamento de tarefas

- Escalonador de tarefas ou *scheduler*
  - *Responsável pela gestão do processador.*
  - *O escalonador faz a escolha de qual tarefa será executada de acordo com critérios pré-estabelecidos (algoritmos de prioridade, tempo de execução, etc) dentro as tarefas prontas para executar.*
  - *Um processo pode ser retirado de execução para executar outro mais prioritário, de acordo com o método de escalonamento.*
  - *Políticas de escalonamento definem critérios ou regras para a ordenação das tarefas de tempo real.*

Escalonamento de tarefas

- A partir das regras, são geradas as escalas, que caso possam ser implementadas, garantem o cumprimento das restrições temporais impostas às tarefas de tempo real.
- Os algoritmos de escalonamento são classificados em:
  - Estáticos: Quando o cálculo da escala é feito a partir de parâmetros fixos atribuídos às tarefas em tempo de projeto.
  - Dinâmicos: Baseados em parâmetros que variam em tempo de execução com a evolução do sistema.

Escalonamento de tarefas

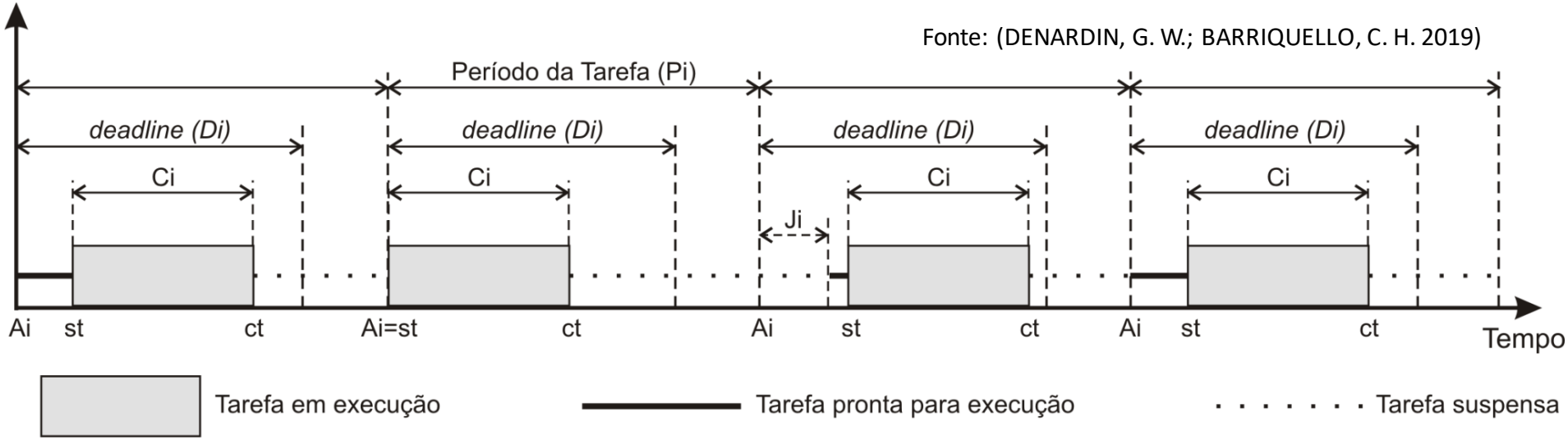
- Quando a escala de tempo é obtida durante o projeto, o algoritmo é *offline*, caso contrário, *online* (produzida em tempo de execução)
- Com isso, podemos ter algoritmos *offline* estáticos, *online* estáticos e *online* dinâmicos.
- Se as tarefas não podem ser interrompidas por outras tarefas, o algoritmo é não preemptivo, caso contrário, é dito preemptivo.

Escalonamento de tarefas

- Uma aplicação apresenta características de carga estática quando todas as suas tarefas são bem conhecidas em tempo de projeto devido às suas restrições temporais (tempo de chegada). Mesmo as situações de pior caso também são conhecidas em tempo de projeto.
- Conhecendo os tempos de chegada, é possível a determinação dos prazos a que uma carga está sujeita.
- Cargas estáticas são modeladas através de tarefas periódicas e esporádicas (sub classe das aperiódicas).
  - A garantia em tempo de projeto para as tarefas aperiódicas pode ser possível assumindo uma taxa máxima de chegada para os eventos críticos.
  - Tarefas esporádicas possuem comportamento determinístico, o que permite a obtenção de garantias em tempo de projeto. Ex. de tarefa esporádica: entrada de um teclado.

Escalonamento de tarefas

Diagrama para tarefas periódicas

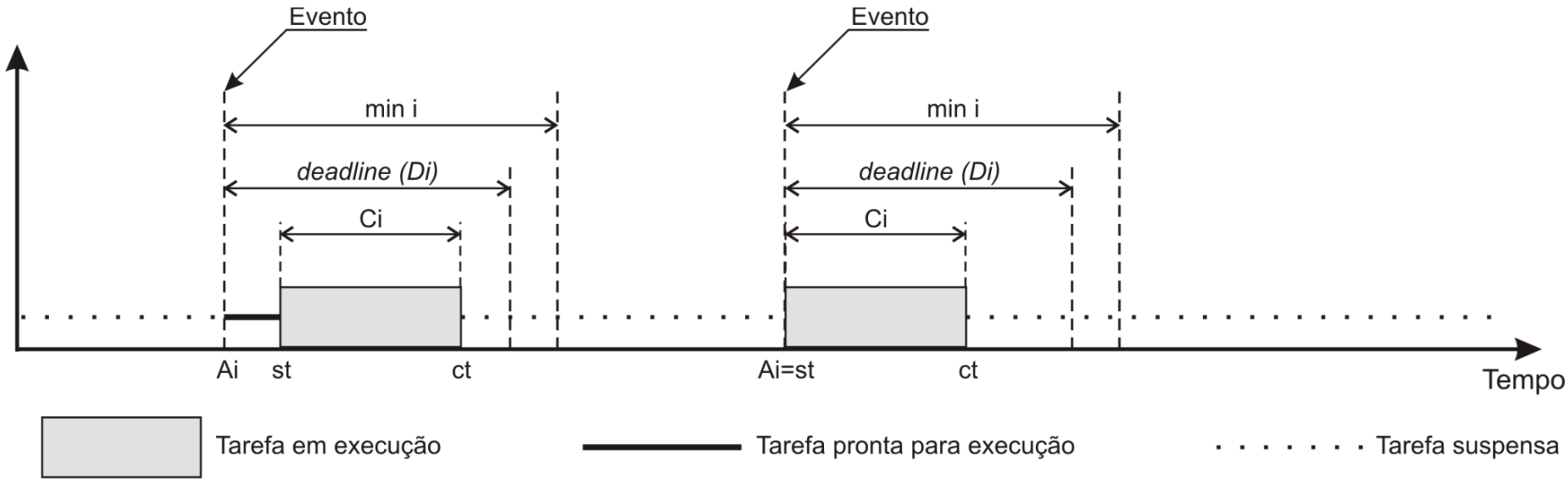


Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. 2019)

Onde *Pi* = período da tarefa, *Ci* = tempo de computação da tarefa, *Di* = Deadline e *Ji* é o Release Jitter

Diagrama para tarefas aperiódicas

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. 2019)



Onde: *Ci* = tempo de computação da tarefa, *Di* = Deadline e *min i* é o mínimo intervalo entre duas requisições consecutivas.

Escalonamento de tarefas

- Despachador: Parte do escalonador responsável por parar a execução de uma tarefa e por outra no lugar.
- Após o escalonador decidir qual tarefa deve ser executada, o despachador salva o conteúdo de todos os registradores daquela tarefa para que ela possa continuar a execução no futuro (troca de contexto).
- Esse gerenciamento pode ser feito na tarefa, ISR ou kernel (executando alguma função do próprio SO).

Escalonamento de tarefas

- Os escalonadores de tarefas mais utilizados são:
  - Escalonamento dirigido por tempo
    - FIFO e SJF
    - Executivo cíclico
    - Round-robin
  - Escalonamento dirigido por prioridades
    - Taxa monotônica - *Rate monotonic – RM*
    - *"Prazo monotônico" e "prazo mais cedo primeiro" (Earliest deadline First – EDF)*

Escalonamento de tarefas

- Executivo cíclico
  - *Implementado em sistemas multitarefas cooperativos dirigidos por tempo. É determinístico, com construção da escala de execução estática e offline.*
  - *Todo o trabalho de escalonamento é feito em projeto. Resulta em uma grade de execução.*
  - *A ideia principal é executar tarefas de forma cíclica, com frequência bem definida. A execução se resume a um simples despachante, ativando ciclicamente tarefas segundo uma escala.*
  - *A escala calculada sempre reflete o pior caso, o que gera desperdício de recursos.*
  - *Vantagem: Comportamento totalmente conhecido*

Escalonamento de tarefas

- Executivo cíclico: Define 2 intervalos de tempo
  - *Tempo de ciclo principal (TCP): dado pelo mínimo múltiplo comum dos períodos das tarefas e, deve conter um número inteiro de tempos de ciclo secundário.*
  - *Tempo de ciclo secundário (TCS): deve satisfazer a seguinte relação para cada tarefa do sistema:*

$$TCS + (TCS - máximo\ divisor\ comum(TCS, Pi)) \leq Di$$

Onde Pi e Di são o período e o prazo (deadline) de uma determinada tarefa

Escalonamento de tarefas

Tarefas	Tempo de computação (Ci)	Período da tarefa (Pi)
A	10	25
B	8	25
C	5	50
D	4	50
E	2	100

Cálculo de TCP:

$$TCP = mínimo\ multiplo\ comum\ (P1, P2, P3, P4, P5)$$

$$TCP = mínimo\ multiplo\ comum\ (25,25,50,50,100) = 100$$

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. 2019)

Escalonamento de tarefas

Tarefas	Tempo de computação (Ci)	Período da tarefa (Pi)
A	10	25
B	8	25
C	5	50
D	4	50
E	2	100

- Cálculo de TCS

$$TCS + (TCS - máximo\ divisor\ comum(TCS, Pi)) \leq Di$$

Onde Pi e Di são o período e o prazo (deadline) de uma determinada tarefa. O TCS deve ser ≥ ao maior tempo de computação (10) e ≤ ao menor período das tarefas (25).

De início, podemos supor TCS = 25 que é igual ao menor período e divisível por TCP. Com isso, tem-se:

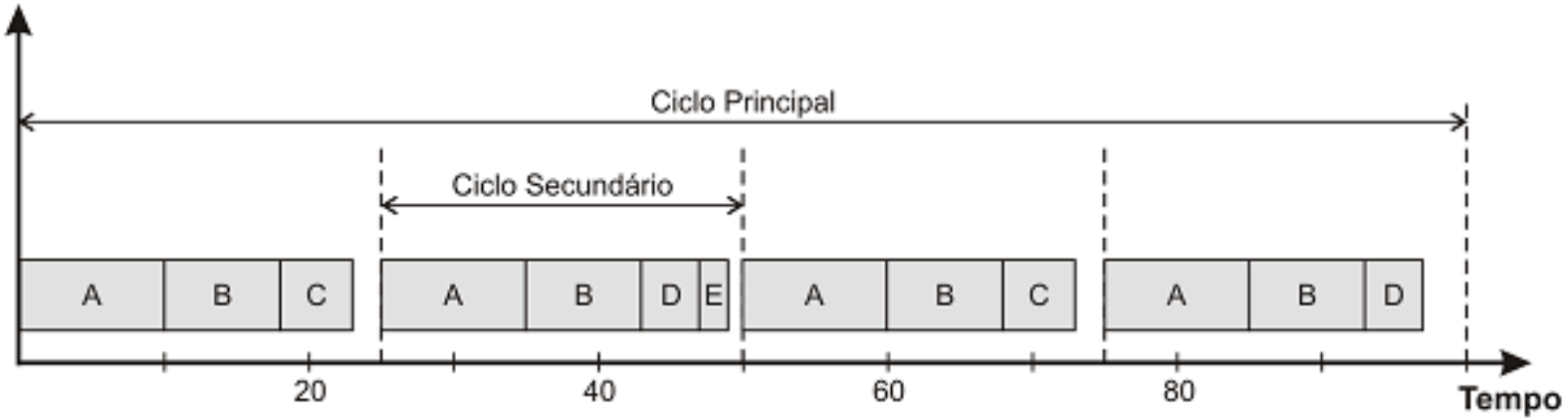
$$50 - máximo\ divisor\ comum\ (25, Pi) \leq Di$$

$$50 - 25 \leq 25, para\ qualquer\ tarefa$$

Escalonamento de tarefas

Tarefas	Tempo de computação (Ci)	Período da tarefa (Pi)
A	10	25
B	8	25
C	5	50
D	4	50
E	2	100

Com isso, tempo que o conjunto de tarefas é escalonável com TCP = 100 e TCS = 25



Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. 2019)

Escalonamento de tarefas

- *Escalonamento dirigido por prioridades*
  - *A partir do nível de prioridade, é determinado o grau de importância de uma determinada tarefa em um sistema operacional.*
  - *As prioridade podem ser estáticas (não se alteram durante a execução da aplicação) ou dinâmicas.*
  - *Caso existam mais de uma tarefa com a mesma prioridade, o escalonador irá selecionar a próxima tarefa a ser executada segundo uma política, que pode ser Round-Robin, FIFO ou SJF (shortest job first).*

Escalonamento de tarefas

- *Escalonamento dirigido por prioridades*
  - *Um problema quando se utiliza prioridade estática, é que uma tarefa de baixa prioridade poderá nunca ser executada (starvation), ou irá demorar muito mais do que o previsto. Porém, existem técnicas para evitar esse comportamento.*
    - *No sistema de envelhecimento é definida uma metodologia de análise de inanição de tarefas, que pode levar em consideração o tempo, número de preempções, etc. Caso a análise exceda o limite estipulado (exemplo: demorou mais que 500ms), a prioridade da tarefa é alterada para que ela possa competir com as tarefas de maior prioridade.*
    - *No sistema baseado em uso de CPU, uma tarefa com baixa prioridade, porém com uso de CPU menor que as outras tarefas, tem sua prioridade aumentada.*

Escalonamento de tarefas

- *Escalonamento por taxa monotônica - Rate monotonic – RM*
  - Quanto menor o período de execução atribuído a tarefa, maior a prioridade.
  - É a técnica mais utilizada nos RTOS comerciais. Devido a sua simplicidade e por ter resposta ótima dentro das hipóteses consideradas, com prioridades fixas.
    - As tarefas forem periódica e independentes (não compartilham recursos)
    - Seus tempos de computação forem conhecidos e constantes
    - Seus prazos coincidirem com seus períodos
    - Os tempos de troca entre tarefas forem desprezíveis
    - As tarefas não periódicas, caso existirem, não possuírem prazos de execução críticos (ou seja, não são tarefas de tempo real)

Escalonamento de tarefas

- *Escalonamento por taxa monotônica - Rate monotonic – RM*
  - No RM, as prioridades aumentam em função da diminuição dos períodos das tarefas, ou seja, quanto mais frequente a tarefa, maior a sua prioridade no conjunto.
  - Como o período das tarefas é fixo, o RM também define prioridades fixas (estáticas).

Escalonamento de tarefas

- *Escalonamento por taxa monotônica - Rate monotonic – RM*
  - A análise de escalabilidade no RM, feita em tempo de projeto, é baseada no cálculo da utilização (U).
  - Para que n tarefas tenham o atendimento de suas restrições temporais quando escalonadas pelo RM, deve ser satisfeito o teste abaixo que define uma condição suficiente:

$$U = \sum_{i=1}^{n} \frac{Ci}{Pi} \leq n\left(2^{\frac{1}{n}} - 1\right)$$

  - A medida que n cresce, nesse teste, a utilização do processador converge para 0,69. Uma utilização de aproximadamente 70% define uma baixa ocupação do processador que, certamente, implica no descarte de muitos conjuntos de tarefas com utilização maior e que, mesmo assim, apresentam escalas realizáveis.
  - A condição necessária pode ser relaxada quando as tarefas tiverem períodos múltiplos da tarefa mais prioritária.

Escalonamento de tarefas

- *Escalonamento por taxa monotônica - Rate monotonic – RM*
  - *Exemplo: Verificar a escalonabilidade desse conjunto de tarefas*

Tarefas Periódicas	Período das tarefas (P_i)	Tempo de Computação (C_i)	Prioridade RM	Utilização (U_i)
Tarefa A	80	30	1	0,375
Tarefa B	40	5	2	0,125
Tarefa C	16	4	3	0,250

  - *Aplicando a equação anterior, temos que:*

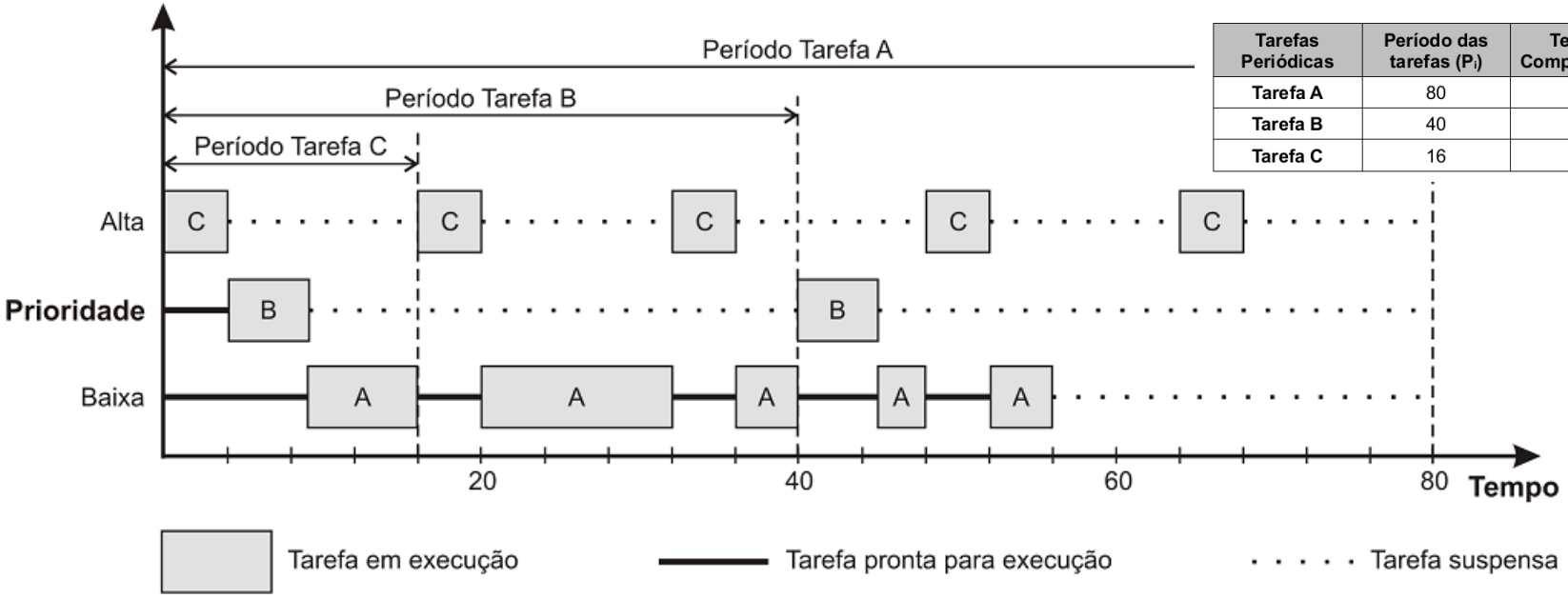
$$U = \sum_{i=1}^{n} \frac{Ci}{Pi} \leq n\left(2^{\frac{1}{n}} - 1\right) \qquad 0{,}75 \leq n\left(2^{\frac{1}{n}} - 1\right) = 0{,}779$$

  - *O que nos mostra que o conjunto de tarefas pode ser escalonável por RM*

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. . Pg. 144, 2019)

Escalonamento de tarefas

- *Na imagem abaixo, temos o escalonamento das tarefas*



Tarefas Periódicas	Período das tarefas (P_i)	Tempo de Computação (C_i)	Prioridade RM
Tarefa A	80	30	1
Tarefa B	40	5	2
Tarefa C	16	4	3

- *Por ser mais frequente, a tarefa C tem maior prioridade pelo RM e assume o controle da CPU. Em t=4, a tarefa C conclui e B toma posse do processador por ser a mais prioritária na fila. A tarefa A assume o processador em t=9 e é interrompida quando ocorre nova ativação da tarefa C em t=16 (inicia um novo período da tarefa C), a qual passa a ser executada, e assim sucessivamente. Pode-se notar que foi utilizado apenas 75% do tempo de processamento, sendo que os outros 25% podem ser utilizados na implementação de rotinas de aquisição de informações de execução das tarefas, baixo consumo de energia, etc.*

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. . Pg. 144, 2019)

Escalonamento de tarefas

- *Escalonamento por taxa monotônica - Rate monotonic – RM*
  - *Exemplo 2: Verificar a escalonabilidade desse conjunto de tarefas*

Tarefas Periódicas	Período das tarefas (P_i)	Tempo de Computação (C_i)	Prioridade RM	Utilização (U_i)
Tarefa A	50	12	1	0,240
Tarefa B	40	10	2	0,250
Tarefa C	30	10	3	0,333

  - *Aplicando a equação, temos que:*

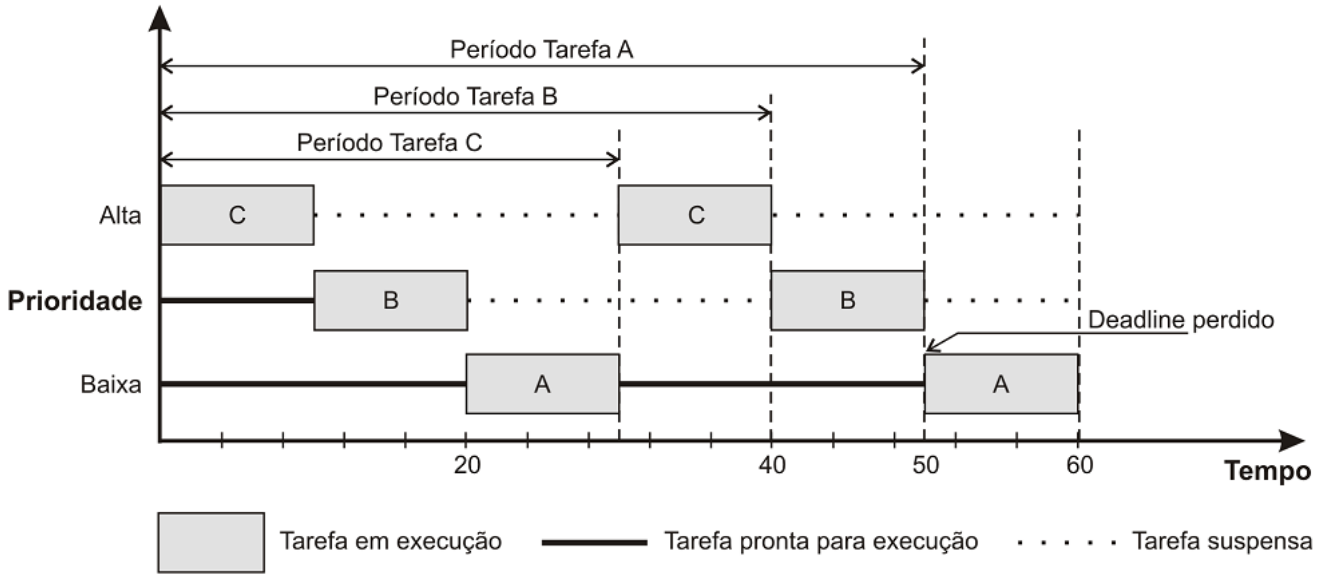
$$U = \sum_{i=1}^{n} \frac{Ci}{Pi} \leq n\left(2^{\frac{1}{n}} - 1\right) \qquad 0{,}823 > n\left(2^{\frac{1}{n}} - 1\right) = 0{,}779$$

  - *O que nos mostra que o conjunto de tarefas não pode ser escalonável por RM.*

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. . Pg. 144, 2019)

Escalonamento de tarefas

Tarefas Periódicas	Período das tarefas (P_i)	Tempo de Computação (C_i)	Prioridade RM	Utilização (U_i)
Tarefa A	50	12	1	0,240
Tarefa B	40	10	2	0,250
Tarefa C	30	10	3	0,333



- *Conforme podemos ver na imagem acima, não foi possível atender o tempo de computação da tarefa A dentro do seu período estipulado.*

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. . Pg. 144, 2019)

Escalonamento de tarefas

- *Escalonamento por prazo monotônico (DM)*
  - *No DM as prioridades crescem em função da redução dos prazos (deadlines), ou seja, quanto mais rápido a tarefa deve ser completada, maior a sua prioridade no conjunto.*
  - *Como os deadlines das tarefas não mudam, o DM define uma atribuição estática de prioridades (prioridade fixa).*
  - *O algoritmo DM apresenta uma vantagem em relação ao RM, que é o gerenciamento de tarefas aperiódicas, visto que essas tarefas possuem prazos para serem processadas.*

Escalonamento de tarefas

- *Escalonamento por "prazo mais cedo primeiro" - Earliest deadline first – EDF*
  - Algoritmo de escalonamento dinâmico, ou seja, as tarefas possuem prioridades dinâmicas (mudam ao longo do tempo).
  - A prioridade é definida de acordo com o deadline da tarefa, ou seja, quanto menor o deadline, maior a prioridade de execução da tarefa.
  - Entrega um bom resultado em situações que seja necessário gerenciar tarefas aperiódicas, visto que essas tarefas também possuam prazo para serem executadas.

Escalonamento de tarefas

- *Escalonamento FIFO*
  - *Seu funcionamento é utilizando uma fila*
  - *Tarefas prontas para execução, entram no fim da fila*
  - *A tarefa que estiver no início da fila é a próxima a ser executada*
  - *A tarefa é executada até que libere explicitamente a CPU, realize uma chamada do sistema ou termine sua execução.*
    - *Esse algoritmo só é possível de ser utilizado em sistemas cooperativos ou híbridos.*

Escalonamento de tarefas

- *Escalonamento FIFO*
  - *Para calcular o tempo médio de espera de um determinado conjunto de tarefas, soma-se os tempos iniciais em que cada tarefa presente na fila demorou para ser executada e, divide-se o valor desta soma pelo número de tarefas na fila.*

Tarefas	Tempo de computação (Ci)
A	12
B	8
C	15
D	5

Ordem A-B-C-D = $\frac{(0+12+20+35)}{4} = 16,75$

Ordem D-A-B-C = $\frac{(0+5+17+25)}{4} = 11,7$

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. . Pg. 144, 2019)

Escalonamento de tarefas

- *Escalonamento Shortest Job First (SJF)*
  - *Nesse caso, o algoritmo organiza a fila dando prioridade para as tarefas com menores tempos de computação.*
  - *O maior problema é determinar o tempo de execução de cada tarefa, uma vez que as tarefas incluídas na fila podem se alterar dependendo do funcionamento da aplicação.*

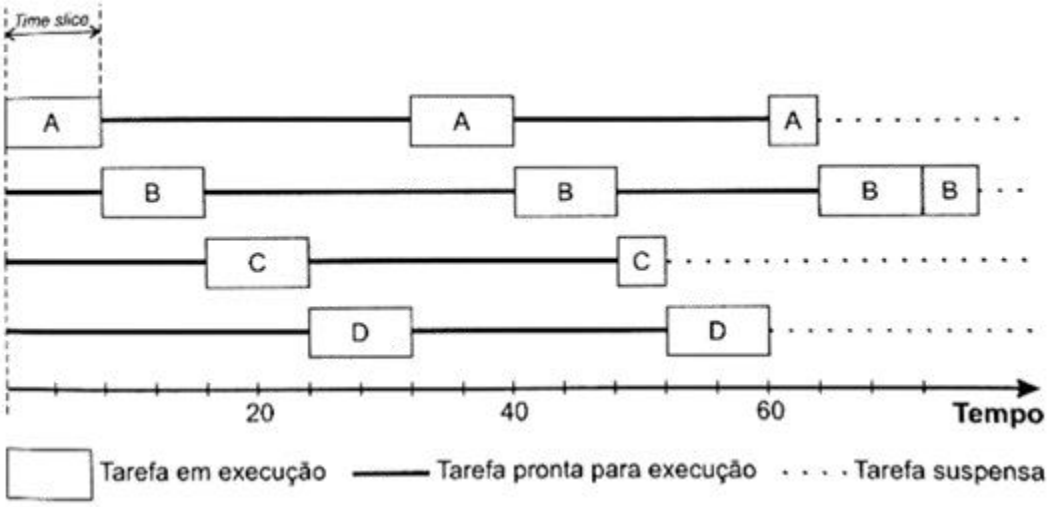
Tarefas	Tempo de computação (Ci)
A	12
B	8
C	15
D	5

Ordem D-B-A-C = $\frac{(0+5+13+25)}{4} = 10,75$

Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. . Pg. 144, 2019)

Escalonamento de tarefas

- *Round Robin ou Time Slicing*
  - *É um dos mais antigos e simples algoritmos de escalonamento. Foi projetado especialmente para sistemas de compartilhamento de tempo, sendo necessário o uso de temporizador.*
  - *É configurado um tempo de processamento fixo para as tarefas (time slice), e o processamento das tarefas ocorrem dentro dessas janelas*
  - *Na figura abaixo, note que uma tarefa perde o controle da CPU quando termina a sua execução ou a sua fatia de tempo é esgotada.*



Fonte: (DENARDIN, G. W.; BARRIQUELLO, C. H. 2019)

Escalonamento de tarefas

- *Notas:*
- *O desempenho de um sistema operacional está diretamente relacionado com seu escalonador de tarefas.*
- *O projeto de um escalonador deve considerar a aplicação fim do sistema.*
  - *Por exemplo, um OS dedicado a experiência do usuário (interface homem/máquina) possui diferentes parâmetros do que um OS dedicado ao controle de processos em tempo real.*
- *A escolha de um escalonador não depende apenas da velocidade em que a próxima tarefa é selecionada. Também deve ser levado em conta o tempo de manutenção da fila já formada com as tarefas prontas.*

Referências

- DENARDIN, G. W.; BARRIQUELLO, C. H. Sistemas Operacionais de Tempo Real e sua Aplicação em Sistemas Embarcados. 1ª edição, São Paulo, Blucher, 2019.
- DENARDIN, G. W. Notas de aula de sistemas embarcados. UTFPR.
- STALLINGS, William. Operating systems: internals and design principles. 5.ed. Upper Saddle River: Pearson Prentice Hall. 2004.
- TANENBAUM, Andrew. Sistemas operacionais modernos. Rio de Janeiro: LTC. 1999.
- BACURAU, R.M. Notas de aulas de projeto de sistemas embarcados. UNICAMP, 2020. Disponível em: https://sites.google.com/site/rodrigobacurau/cursos-2020-1/es670---projeto-de-sistemas-embarcados