

Ch.10 – Multiprocessor and Real-Time Scheduling

10.1 – Multiprocessor Scheduling

10.1.1 Granularity

10.1.2 Design Issues

10.1.3 Process Scheduling

10.1.4 Thread Scheduling

10.2 – Real-Time Scheduling

10.2.1 Background

10.2.2 Characteristics of Real-Time OS

10.2.3 Real-Time Scheduling

10.2.4 Deadline Scheduling

Ch.10 – Multiprocessor and Real-Time Scheduling

10.3 – LINUX Scheduling (LEITURA COMPLEMENTAR)

10.4 – UNIX SVR4 Scheduling (LEITURA COMPLEMENTAR)

10.5 – UNIX FreeBSD Scheduling (LEITURA COMPLEMENTAR)

10.6 – WINDOWS Scheduling (LEITURA COMPLEMENTAR)

Referências Bibliográficas

- Operating Systems – Internals and Design Principles. William Stallings. 7th, Prentice-Hall 2012.
- Instructor Resources – Operating Systems - 7th
<http://williamstallings.com/OperatingSystems/OS7e-Instructor/>

Bear in mind, Sir Henry, one of the phrases in that queer old legend which Dr. Mortimer has read to us, and avoid the moor in those hours of darkness when the powers of evil are exalted.

THE HOUND OF THE BASKERVILLES, Arthur Conan Doyle

10. Multiprocessor and Real-Time Scheduling

- “**multiprocessor**” .. quando um sistema computacional contempla mais de 01 processador, requisitos funcionais adicionais são necessários no projeto das funções de escalonamento.
- ... análise semelhante cabe aos sistemas de tempo real, que além dos requisitos tradicionais normalmente contemplados em escalonadores, atrelam as ações na linha do tempo.
- Esta discussão está dividida em 02 grandes seções, quais sejam, “real-time scheduling” e “multiprocessor scheduling”.
- ... no “livro texto” são também apresentados alguns aspectos de projeto de escalonador de sistemas operacionais contemporâneos como o UNIX, Linux e Windows (Leitura Complementar).

10. Multiprocessor and Real-Time Scheduling

10.1 - Multiprocessor Scheduling

- **“classificação”** .. sistemas multiprocessados ..
- **“fracamente acoplados”** .. coleção de máquinas autônomas, cada qual com o seu próprio processador e memória principal bem como canal de entrada/saída .. “Multiprocessadores Distribuídos” ou “Clusters”.
- **“fortemente acoplados”** .. coleção de processadores que compartilham um memória principal e sobre a qual são integrados e controlados por um sistema operacional.
- **“especializados funcionalmente”** .. e.g. processador de entrada/saída, onde um processador de propósito geral ou processador mestre controla processadores específicos que oferecem serviços.
- **objetivo** .. discutir as características dos Sistemas Multiprocessados Fortemente Acoplados, principalmente as características relacionadas ao escalonador de processos.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

10.1.1 – Frequency of Synchronization

- “**granularidade**” .. também conhecido por granularidade de sincronização ou frequência de sincronização, delimita a frequência com que processos do sistema são sincronizados.
- ... podemos distinguir 05 categorias de paralelismo que contemplam diferentes níveis de granularidade:
- “granularidade fina”; “granularidade média”; “granularidade grosseira”; “granularidade muito grosseira” e “paralelismo independente”.
- ... cada nível contempla diferentes frequências de sincronização, normalmente medidas em número de instruções.
- ... que podem ser mapeadas para tempo (e.g., 10^{-3} seg., 10^{-9} seg.) considerando os aspectos de organização dos sist. computacionais.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.1 – Frequency of Synchronization

- “**paralelismo independente**” – não há sincronização explícita entre processos, ou seja, cada processo representa uma tarefa.
- ... normalmente aplicado em sistemas de “time-sharing”, e.g., cada usuário realiza um tarefa em particular, como processamento de texto ou manuseio de uma planilha.
- ... neste cenário o multiprocessador provê o mesmo serviço que um sistema monoprocessado multiprogramado, logo, o desempenho é semelhante ao encontrado em “desktops”.
- ... se arquivos ou informação devem ser compartilhados, sistemas individuais devem ser mantidos juntos em um sistema distribuído normalmente suportado por um rede de computadores.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.1 – Frequency of Synchronization

- “**granularidade grosseira ou muito grosseira**” .. há sincronização entre processos, mas em um nível grosseiro, que em instruções representa de 200 a 1.000.000 de instruções.
- ... tais situações são tratadas como processos concorrentes executando em sistemas monoprocessados multiprogramados.
- ... mas podem ser suportados em sistemas multiprocessados com pouca ou nenhuma mudança no software.
- ... em geral, qualquer coleção de processos concorrentes que necessite se comunicar pode se beneficiar do uso de sistemas multiprocessados.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.1 – Frequency of Synchronization

- e.g., nos casos de processos cuja interação é ocasional, um sistema distribuído pode proporcionar um bom suporte.
- ... por outro lado, se os processos interagem com frequência, a sobrecarga da comunicação pela rede pode contrapor o aumento potencial de velocidade.
- ... nestes casos, o multiprocessador provê suporte mais efetivo quando comparado a outras organizações de máquina.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.1 – Frequency of Synchronization

- “**granularidade média**” .. como discutido no Cap. 04, constitui-se uma coleção de “threads” dentro de um único processo.
- ... programadores devem explicitar o paralelismo potencial de uma aplicação, exigindo assim um grau maior de coordenação e interação entre as “threads” de uma mesma aplicação.
- ... normalmente contempla frequência de sincronização entre 20 e 200 instruções de máquina (sincronização dentro do processo).
- ... no paralelismo com granularidade média é necessário reexaminar o escalonamento em razão da presença das “threads”.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.1 – Synchronization Granularity

- “**granularidade média**” .. diferente do “paralelismo independente”, “grosseiro” ou “muito grosseiro” é necessário reexaminar o agendamento ao negociar com o escalonamento de threads.
- .. pouco ou nenhum impacto está presente no caso do “paralelismo independente”, “muito grosseiro” e “grosseiro” suportado em sistemas multiprocessado multiprogramado ou em um multiprocessador .
- ... em função da frequência de interação entre “threads” de uma aplicação, decisões de escalonamento relacionadas a uma “thread” podem afetar o desempenho de toda a aplicação.

10. SMP and Real-Time Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.1 – Synchronization Granularity

- “**granularidade fina**” .. contempla o uso mais complexo do paralelismo do que o encontrado entre “threads”, ou seja, paralelismo inerente ao fluxo de instruções – único fluxo de inst.
- .. embora muito trabalho tenha sido feito no domínio de aplicações paralelas, esta é uma área muito específica e fragmentada, onde diferentes abordagens estão presentes.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

10.1.2 – Design Issues

- Escalonamento em Multiprocessadores envolve 03 aspectos:
 - .. atribuição de processos a processadores;
 - .. uso de processador individual ou da multiprogramação;
 - .. instanciação/expedição/envio real de um processo.
-
- Obs.: ... uso da abordagem irá depender em geral da granularidade do paralelismo e do nro. de processos disponíveis.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- “**atribuição de processos a processadores**” .. na forma mais simples, cada processador será tratado com recurso agrupado e o escalonador atribuirá sob demanda processos aos processadores;
- ... desde que se assuma que a arquitetura do multiprocessador é uniforme, no sentido de que nenhum processador tenha vantagem particular no acesso à memória ou dispositivo de entrada/saída.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- ... se um processo é permanentemente atribuído a um processador durante todo o ciclo de vida, então uma fila de baixa periodicidade deve ser mantida para cada processador;
- “**vantagem**” .. “overhead” menor na função de escalonamento, pois a atribuição do processador é mantida por um longo período de tempo para cada processo.
- ... além disso, o uso de processadores dedicados permite o uso da estratégia conhecida com “gang” ou “group scheduling”.
- “**desvantagem**” .. processador com um acúmulo de processos na fila enquanto outro processador está coma fila vazia.
- “**alternativa**” .. balanceamento dinâmico de carga, onde “threads” podem ser deslocadas de um processador para a fila de outro.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- ... independente do processo manter-se associado ou não ao mesmo processador, faz-se necessário algum mecanismo para associação/atribuição de processo a processador:
- “**master/slave architecture**” .. funções chaves do “kernel” do sistema operacional são executadas no processador em particular e os demais processadores executam programas do usuário.
- “**peer architecture**” .. “kernel” pode ser executado em qualquer processador e cada processador efetua o auto escalonamento do “pool” de processos que estão disponíveis para serem executados.
- Abordagem “master/slave” apresenta 02 desvantagens:
- falha no processador mestre leva todo o sistema ao colapso;
- processador mestre pode se tornar o gargalo no desempenho.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- Já a abordagem “peer” torna mais complexo o sistema operacional, pois o mesmo tem que garantir:
- ... 02 processadores não escolham o mesmo processo;
- ... processos não se encontrem de alguma forma perdidos nas filas.
- Há, naturalmente, abordagens entre estes 02 extremos;
- e.g., subconjunto de processadores dedicados para o “kernel” ao invés de um único processador, ou ..
- .. gerenciamento das diferentes necessidades do “kernel” e dos demais processos tendo por base a histórico de execução.
- .. gerenciamento das diferentes necessidades do “kernel” e dos demais processos tendo por base tempo médio de execução dos processos.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- **“uso de processadores individuais ou multiprogramados”** - se um processo está estaticamente associado a um processador, surge a questão - processador deve ser multiprogramado !?
- ... questão procede pois no caso de processos que estão frequentemente bloqueados por entrada/saída ou concorrendo por recursos, o desempenho do processador é baixo.
- ... em sistemas multiprocessados tradicionais, nos quais lida-se com “granularidade grosseira” ou “independente”, é claro que cada processador deve ser capaz de escalonar processos.
- .. para atingir maior utilização e por consequência melhor desempenho.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- Para aplicações com “granularidade média” em um sistema multiprocessado com muitos processadores, não é surpresa que cada processador esteja ocupado tanto quanto possível.
- ... não obstante, o que se deseja é oferecer o melhor desempenho, na média, para as aplicações.
- ... nestes casos, uma aplicação que consiste de inúmeras “threads”, a mesma é executada com baixo desempenho a menos as “threads” possam ser executadas simultaneamente.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.2 – Design Issues

- “**expedição/escolha real de um processo**” - ... escolha dentre os processos disponíveis ou prontos, qual será escalonado !?
- ... em sistemas monoprocessados multiprogramados, o uso de primitivas ou algoritmos de escalonamento sofisticados baseado no histórico de uso pode melhorar o desempenho sobre FCFS;
- ... em sistemas multiprocessados estas complexidades não são necessárias ou são contra-produtivas.
- ... assim, uma abordagem mais simples pode ser mais efetiva por acomodar um menor “**overhead**”.
- No caso de escalonamento de “threads”, questões novas podem ser mais importantes que a prioridade ou histórico de execução.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

10.1.3 – Process Scheduling

- “**escalonamento de processo**” .. determina dentre os processo prontos para serem executados, qual deve ser escalonado.
- .. quando consideramos sistemas multiprocessados, políticas de escalonamento mais simples são mais efetivas por possuírem baixo “**overhead**” - e.g., escalonamento de “threads”.
- .. seja o caso de uma única fila para todos os processadores, ou algum tipo de prioridade com ou sem múltiplas filas, pode-se ver o sistema como um uma arquitetura multiservidor de filas.

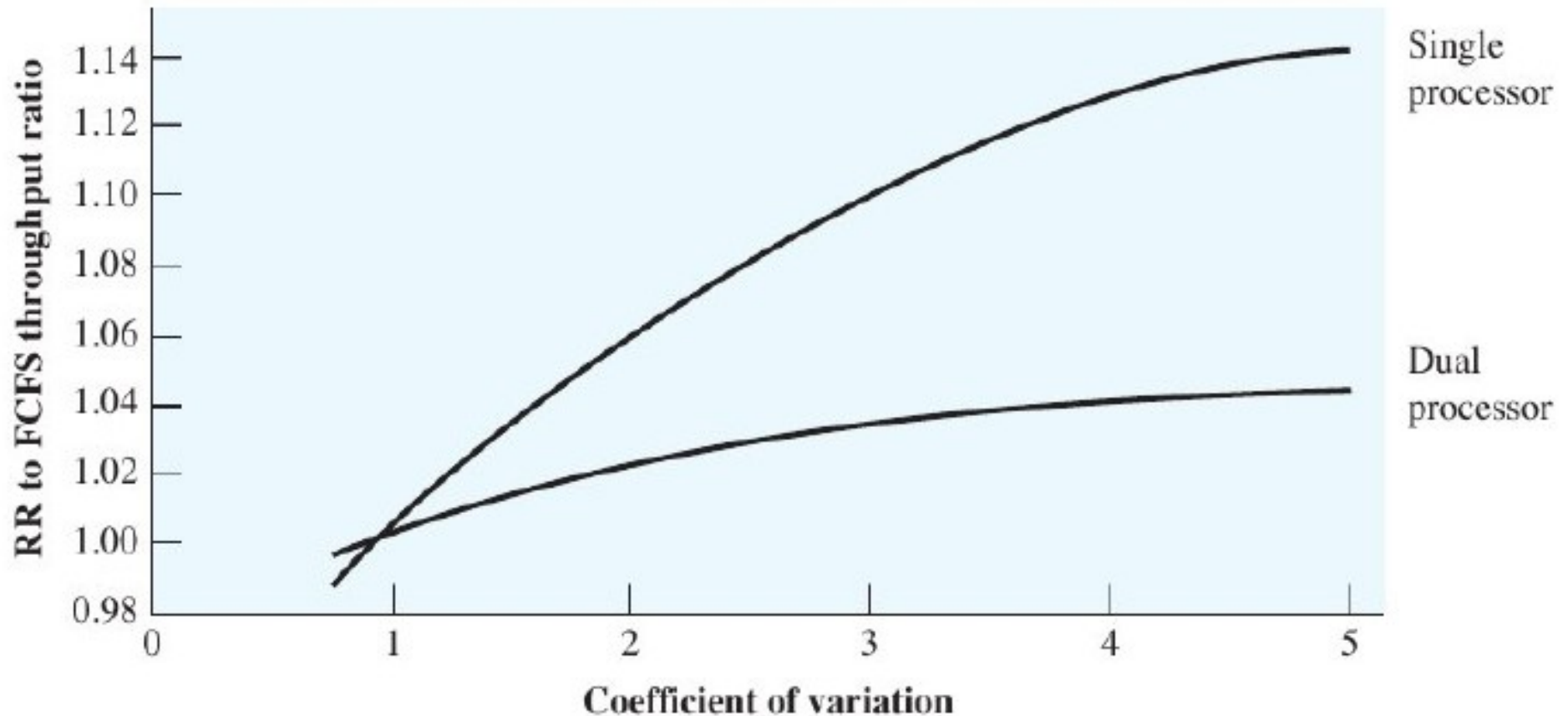
10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.3 – Process Scheduling

- e.g., considere o caso de um sistema com 02 processadores onde cada processador tenha $1/2$ da taxa de um processador comparado a um sistema com um único processador.
- ... compara-se o FCFS com Round Robin e SRT do ponto de vista da fila de processos, bem como o tempo de serviço dos processos.
- ... no caso do “round robin”, o “**quantum**” de tempo é maior que o “**overhead**” da mudança de contexto e menor comparado com o tempo médio de serviço.
- ... tempo de serviço ... mede a parcela de tempo do processador que o processo necessita, quer para toda a tarefa ou para cada momento que o processo está pronto para utilizar o processador.

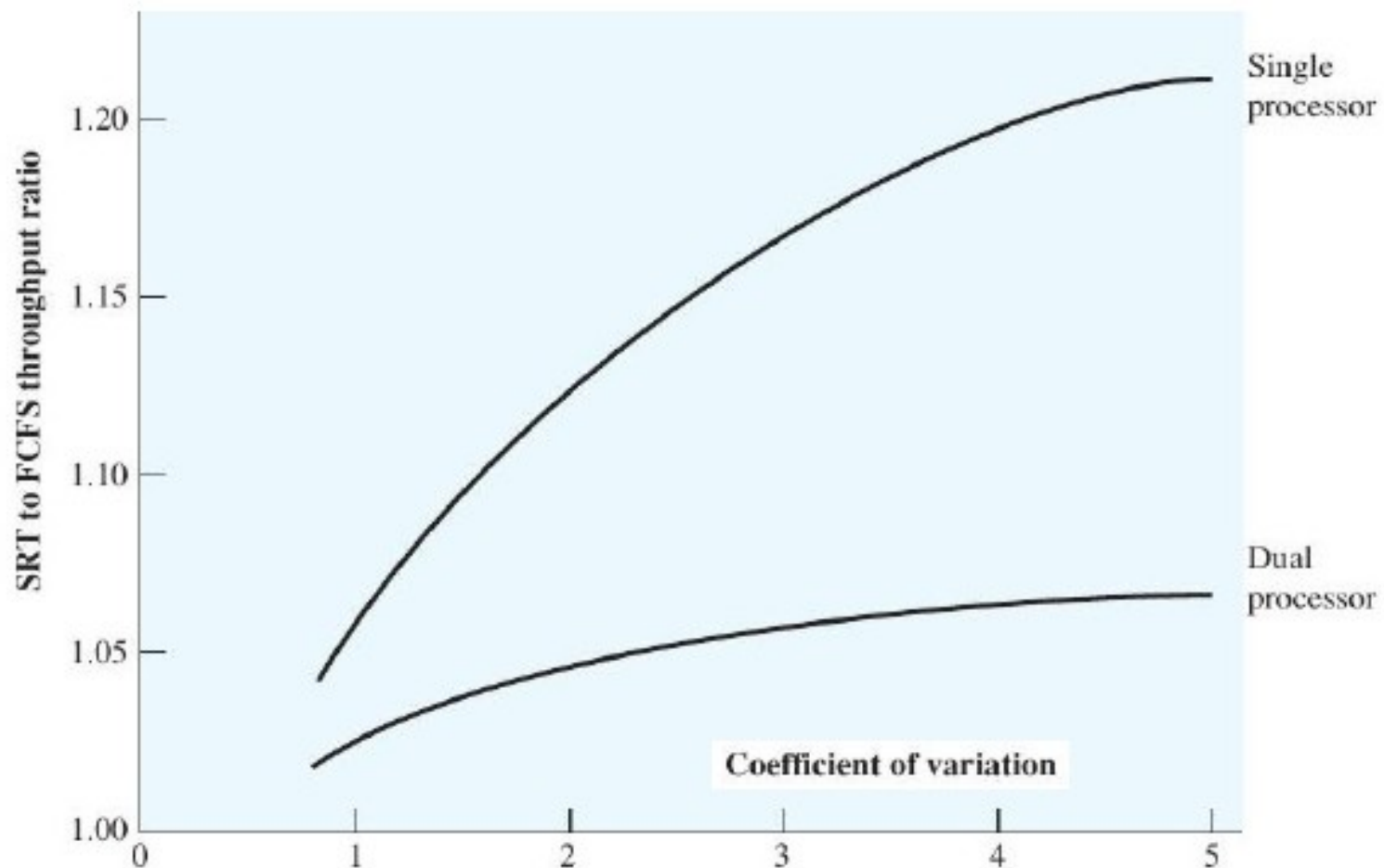
10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling ... 10.1.3 – Process Scheduling

- Fig. 10.2 - Comparação do desempenho de escalonamento para 01 e 02 processadores – Round Robim vs FCFS.



10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling ... 10.1.3 – Process Scheduling

- Fig. 10.2 - Comparação do desempenho de escalonamento para 01 e 02 processadores – SRT vs FCFS.



10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.3 – Process Scheduling

- “**análise**” .. note que a diferença entre os algoritmos de escalonamento é menor no caso de 02 processadores – processo único com um tempo de serviço longo é menos disruptivo no FCFS.
- ... testes (SAUE 1981) foram repetidos sob inúmeras suposições de grau de multiprogramação, mistura de processos “CPU Bound” e “I/O Bound” bem como uso de prioridade.
- “**conclusão**” .. disciplina de escalonamento é menos importante com 02 processadores que com 01 processador, a medida que o nro. de processadores cresce na análise.
- .. assim, a disciplina FCFS ou o uso do FCFS com esquema de prioridade é suficiente para um sistema multiprocessado.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

10.1.4 – Thread Scheduling

- **“escalonamento de threads”** .. como já visto, o conceito de “thread” é o de unidade escalonável de um processo, logo, ..
- ... uma aplicação pode ser concebida como um conjunto de “threads” que cooperam e executam concorrentemente no mesmo espaço.
- ... em sistemas monoprocessados, “threads” podem ser usadas como ferramenta adicional para estruturação bem como para sobreposição de processamento e entrada/saída.
- ... em função do “overhead” do chaveamento de “threads” ser mínimo comparado com o chaveamento de processos, os benefícios são obtidos a baixo custo.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- ... entretanto, o poder completo das “threads” torna-se evidente em sistemas multiprocessados, pois podem ser usadas para explorar o verdadeiro paralelismo das aplicações.
- ... se várias “threads” de uma aplicação são executadas simultaneamente em diferentes processadores, ganhos consideráveis são possíveis no desempenho da aplicação.
- ... diferenças sutis no escalonamento e gerenciamento das “threads” podem impactar significativamente no desempenho.
- “**contexto**” .. aplicações que requerem alta interação entre “threads” ou paralelismo de granularidade média.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- “**multiprocessor thread scheduling**” .. dentre as várias propostas para atribuição de processador e escalonamento de “threads” em sistemas multiprocessados, 04 (quatro) abordagens se destacam-se:
- “**load sharing**” .. mantém-se uma fila global para “threads” e cada processador quando ocioso, seleciona uma “thread” da fila.
- “**gang scheduling**” .. conjunto de “threads” relacionadas é escalonado para ser executado em um conjunto de processadores ao mesmo tempo no esquema um por um.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- **“dedicated processor”** .. contempla escalonamento implícito tendo por base as associações de “threads” aos processadores.
- ... para cada processo é alocado um nro de processadores igual ao nro. de “threads” contempladas no processo e pelo tempo de ciclo de vida do processo quando instanciado.
- ... quando o programa é finalizado, os procesadores retornam para o grupo geral para possível alocação por outro processo.
- **“dynamic scheduling”** .. nro de “threads” por processo pode ser alterado no curso de seu ciclo de vida.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- **“Load Sharing”** .. é provavelmente a mais simples e está presente em sistemas monoprocessados, em razão das seguintes vantagens:
- ... carga distribuída uniformemente entre os processadores, garantindo que nenhum processador está ocioso enquanto há trabalho disponível para ser completado.
- ... quando um processo torna-se disponível, a rotina de escalonamento é executada em um processador para selecionar a próxima “thread”, ou seja, ausência de escalonador centralizado.
- .. fila global pode ser mantida usando qualquer dos algoritmos discutidos no Cap. 09, incluindo algoritmo baseado em prioridade bem como algoritmos que consideram a histórico de execução.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- Leut 1990 .. analisou 03 diferentes versões de “load sharing”:
- **“First Come First Served”** .. para cada programa, cada uma das “threads” é colocada no final da fila, assim, a próxima “thread” é escolhida e executada até se completar ou se bloquear.
- **“Smallest Number of Threads First”** .. fila de “threads” como um fila com prioridades, sendo que a prioridade maior é concedida às threads de processos com o menor nro de “threads” não escalonadas.
- **“Preemptive Smallest Number of Threads First”** .. versão preemptiva do “Smallest Number of Threads First”.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- Usando modelos de simulação, constatou-se que para um vasto conjunto de características de programas, o FCFS é superior as outras 02 políticas apresentadas na lista precedente;
- ... adicionalmente, há evidências de que o “**gang scheduling**” é superior ao “**load sharing**”.
- “**principais desvantagens**” do “**Load Sharing**”
- ... fila global ocupa uma região da memória para a qual deve se garantir a exclusão mútua, o que é um gargalo em muitos processadores que querem trabalhar em paralelo.
- ... este problema não é evidente para poucos procesadores mas torna-se evidente para 10s (dezenas) de processadores.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- (cont.) “**principais desvantagens**” do “**Load Sharing**”
- ... “threads” preemptadas são improváveis de serem executadas no mesmo processador e, considerando que cada processador acomoda uma cache local, seu uso é ineficiente.
- ... se todas as “threads” forem tratadas em um grupo comum de “threads”, é improvável que todas as “threads” de um programa consigam acesso aos processadores ao mesmo tempo.
- ... se um alto grau de coordenação for exigido entre as “threads” de um mesmo programa, o chaveamento de processo pode comprometer seriamente o desempenho do conjunto.
- “**observação**” .. apesar das desvantagens potenciais, FCFS é um dos esquemas mais utilizados em sistemas multiprocessados.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- **“Gang Scheduling”** - conceito de escalonamento simultâneo de um conjunto de processos em um conjunto de processadores antecede o uso das “threads” e tem como benefícios:
- .. se processos fortemente relacionados executam em paralelo, o bloqueio de sincronização pode ser reduzido, assim, menor será o nro de chaveamentos de procesos e o desempenho é melhor.
- “overhead” do escalonamento pode ser reduzido pois uma decisão simples afeta inúmeros processadores e processos ao mesmo tempo.
- .. “gang scheduling” pode ser aplicado para escalonamento simultâneo de “threads” que compõem um único processo, sendo usado para aplicações paralelas de granularidade fina ou média;
- **“observação”** .. nestas aplicações o desempenho degrada muito quando qualquer parte da aplicação não está sendo executada enquanto outras partes estão prontas para serem executadas.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- Necessidade do “gang scheduling” é largamente reconhecida e implementações estão presentes em uma variedade de sistemas.
- ... situação óbvia em que há melhora do desempenho de 01 aplicação única é o fato de minimizar o nro. de chaveamentos de processos.
- e.g., ... assuma que uma “thread” de um processo está executando e que num dado momento alcança um ponto de sincronização com uma outra “thread” do mesmo processo.
- ... se a outra “thread” não está executando, mas pronta na fila, a primeira será desligada até o chaveamento do processo ser feito para algum outro processador para trazer a “thread” necessária.
- ... em aplicações com coordenação precisa entre “threads”, tais chaveamentos irão degradar drasticamente o desempenho.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

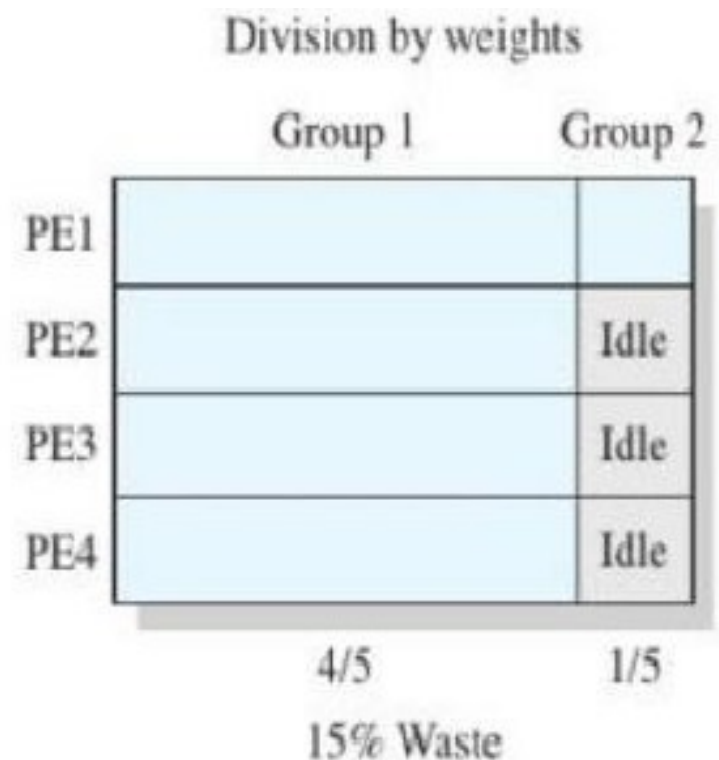
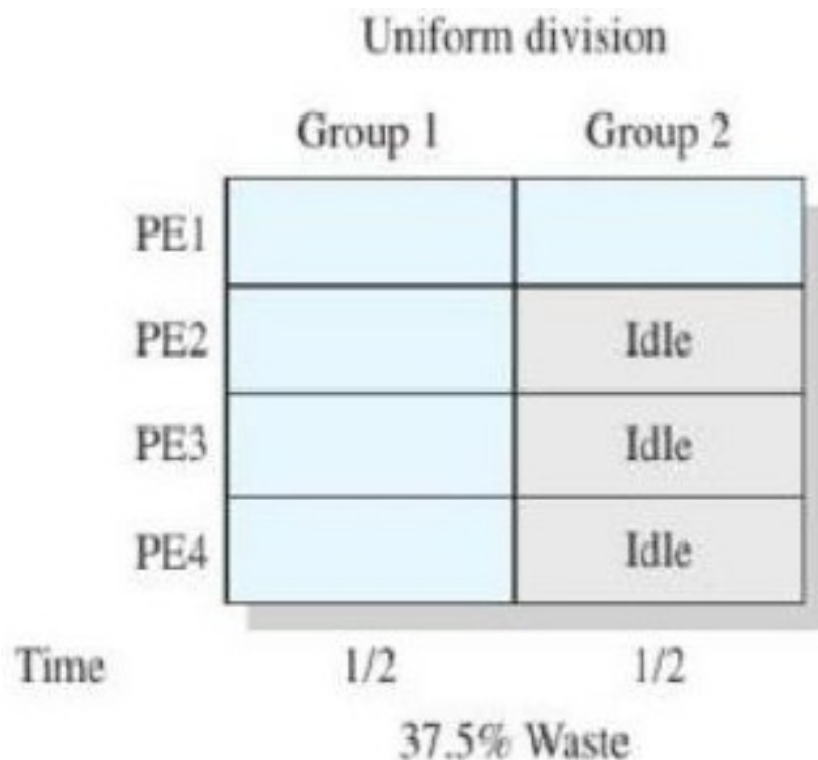
... 10.1.4 – Thread Scheduling

- Suponha que temos “N” processadores e “M” aplicações, cada qual com “N” ou menos “threads”, então a cada aplicação pode ser concedido “1/M” do tempo disponível do “N” processadores.
- ... utilizando “time slicing”, esta estratégia é ineficiente.
- e.g., ... considere 02 aplicações, uma com 04 “threads” e outra com 01 “thread” onde inicialmente a alocação de tempo é uniforme.
- ... neste caso, 37,5% do recurso de processamento será desperdiçado, pois quando a aplicação de “thread” única é executada, 03 processadores ficam ociosos, ou seja, sem atribuição de tarefa.
- ... se tivermos muitas aplicações de uma única “thread”, podemos ajustá-las para aumentar a utilização do processador ou, se não for possível, a ponderação na concessão do tempo é mais eficiente.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

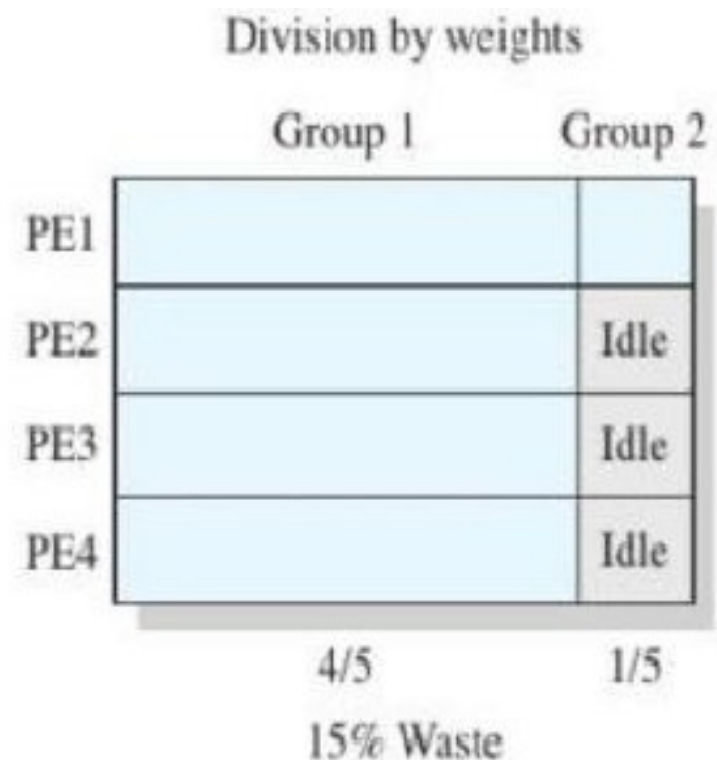
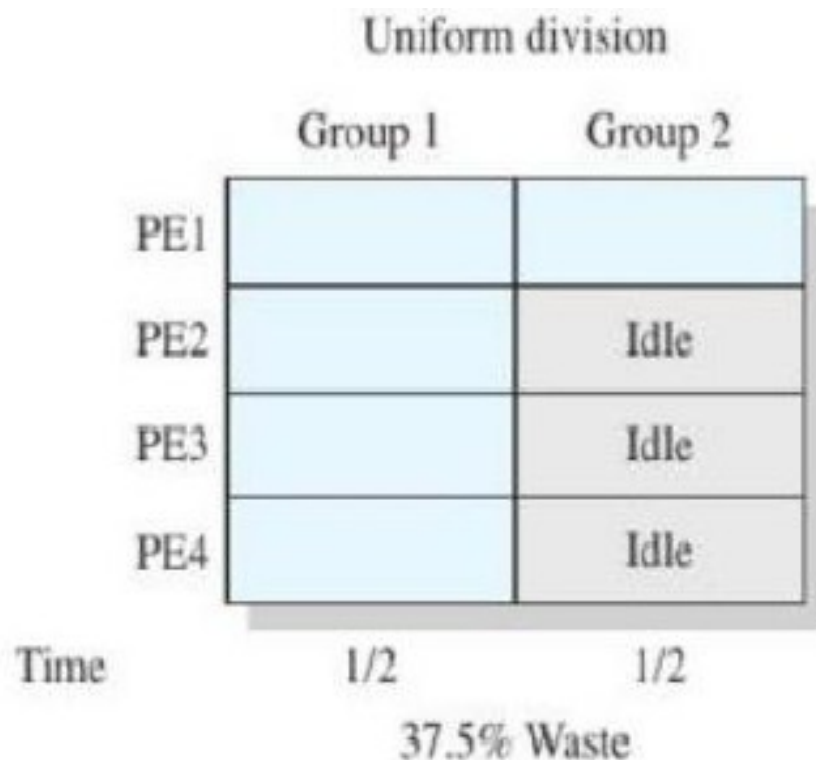
- e.g., .. considere 02 aplicações, uma com 04 “threads” e outra com 01 onde inicialmente a alocação de tempo é uniforme.



10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- ... uma alternativa é o escalonamento que pondera o nro de “threads” por programa, assim ao programa com 04 “threads” será concedido “4/5” do tempo, reduzindo o desperdício para 15%.



10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- **“Dedicated Processor Assignment”** - forma extrema do “gang scheduling”, consiste na associação de um grupo de processadores para uma aplicação durante o seu ciclo de vida.
- ... quando uma aplicação é escalonada, cada “thread” é atribuída a um processador que permanece dedicado à “thread” até que a aplicação complete seu ciclo de vida.
- ... esta abordagem parece proporcionar desperdício grande de tempo do processador, pois se uma “thread” é bloqueada em uma operação de entrada/saída ou sincronização com outra “thread” ..
- ... processador da “thread” bloqueada permanece ocioso, ou seja, não há multiprogramação de processadores.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- 02 observações são válidas em defesa desta estratégia:
- 01 .. em sistemas paralelos com 10s ou 100s de processadores, cada um representa uma fração pequena do custo do sistema.
- .. assim, a utilização do processador não é a métrica mais importante para a eficácia e desempenho.
- 02 .. ao evitarmos o chaveamento de processos durante toda o ciclo de vida do processo, é substancial a aceleração do mesmo.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- Tanto a atribuição de processador dedicado quanto o “gang scheduling” tem como princípio a alocação do processador.
- ... problema de alocação de processador em um sistema multiprocessado assemelha mais ao problema de alocação de memória em sistemas monoprocesados que o problema de escalonamento.
- ... questão é quantos processadores atribuir a um programa em um dado momento é análogo a quantos “frames” devem ser concedidos a um processo durante o seu ciclo de vida.
- CEHR 1987 – propõe o termo “**active working set**” como o número mínimo de “threads” que devem ser escalonadas simultaneamente nos processadores para uma aplicação.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- “**Dynamic Scheduling**” .. para algumas aplicações, é possível contemplar na linguagem ou nas ferramentas do sistema meios que possibilitem a alteração dinâmica do nro de threads.
- ... o que por outro lado, pode permitir ao sistema operacional ajustes de carga para melhorar o desempenho.
- Uma abordagem é permitir que tanto o sistema operacional quanto a aplicação se envolvam no escalonamento.
- ... mas esta responsabilidade de escalonamento é primariamente limitada à alocação do processador.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- Procede de acordo com a seguinte regra, quando uma tarefa requer um ou mais processadores:
- 01 .. se há processadores livres, use-os para atender requisições;
- 02 .. se não, verifique se a “job” requisitante é o mais recente, aloque um processador único retirando-se um de algum “job” que correntemente está alocando 02 ou mais processadores.
- 03 .. se alguma porção da requisição não pode ser satisfeita, ela é marcada até que um processador torne-se disponível ou o “job” desfaz a requisição, p.ex., não mais é necessário um processador extra.

10. SMP and RT Scheduling / 10.1. Multiprocessor Scheduling

... 10.1.4 – Thread Scheduling

- Com a liberação de um ou mais processadores:
- 04 .. busque na fila corrente de requisições não satisfeitas por processadores e atribua um processador único para cada “job” presente na lista corrente de “jobs”.
- ... na sequência, busque novamente na lista para alocação dos processadores restantes segundo FCFS.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

10.2.1 – Background

- “**real-time computing**” .. tema é de extrema importância em sistemas operacionais, tem o “escalador” como elemento de destaque.
- e.g., controle de experimentos em laboratório; controle de processos em plantas industriais; telecomunicações; sistemas de controle; etc.
- e.g., veículo autônomo; controle de robôs e juntas elásticas; sistemas de manufatura inteligente; estação espacial; exploração submarina; etc.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.1 – Background

- “**definição**” .. tipo de computação no qual a corretude do sistema depende não apenas da lógica resultante da computação, mas também do tempo no qual os resultados são produzidos.
- ... podemos definir “sistema de tempo real” definindo o que entende-se por “processos / tarefas de tempo real”.
- “**real-time tasks**” .. tarefas tentam controlar ou reagir a eventos que acontecem fora do sistema computacional.
- ... como estes eventos ocorrem em tempo de execução ou “tempo real”, um “processo de tempo real” deve ser capaz de tratar os eventos pelos quais é responsável no tempo esperado.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

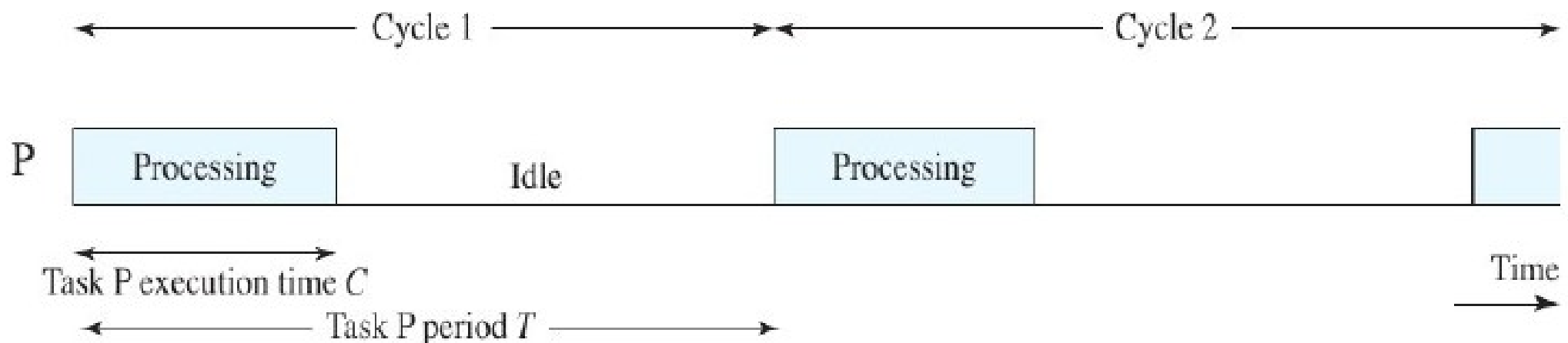
... 10.2.1 – Background

- “**deadline**” .. especifica uma marca de tempo em que uma dada tarefa deve ser iniciada ou finalizada.
- ... tais tarefas podem ser classificadas como “**hard**” ou “**soft**” em função do quão precisas são no tocante a restrição de tempo.
- “**hard real-time task**” .. “deadline” deve ser atendido prontamente, ou então, causará dano inaceitável ou erro fatal.
- “**soft real-time task**” .. atendimento do “deadline” é desejável mas não mandatório, assim, há sentido ao atender a tarefa ainda que o “deadline” não tenha sido satisfeito no tempo esperado.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.1 – Background

- “**periodic / aperiodic**” .. frequência com que uma dada tarefa de “real-time” real deve ser executada.
- “**periodic task**” .. “deadline” deve ser atendido uma vez a cada período de tempo “ T ” ou exatamente a cada “ T ” unidades de tempo.
- “**aperiodic task**” .. “deadline” deve ser atendido uma única vez, seja no início ou no final de uma marca de tempo e sem repetição.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

10.2.2 – Characteristics of Real-Time OS

- Sistema Operacional de Tempo Real podem ser caracterizados por contemplarem requisitos em 05 grandes áreas:
- “**determinístico**” .. sistema operacional é determinístico a medida que atende requisições em tempo fixo ou intervalo de tempo pré-definido, ou seja, mede o atraso desde a sua ocorrência até o reconhecimento.
- “**capacidade de resposta**” .. esta relacionada com quão rápido o sistema operacional atende uma requisição após reconhecê-la.
- “**controle de usuário**” .. essencial que o usuário possa controlar com certo grau de refinamento a prioridade da tarefa.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- Sistema Operacional de Tempo Real podem ser caracterizados por contemplarem requisitos em 05 grandes áreas:
- “**confiabilidade**” .. habilidade do sistema realizar e manter suas funções em circunstâncias rotineiras bem como hostis ou inesperadas.
- “**operação fail-soft**” .. reflete a habilidade do sistema falhar de tal maneira que preserve tanto quanto possível sua capacidade e dados.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**deterministic**” .. sistema operacional é determinístico a medida que realiza/conclui operações em tempo fixo previamente definido ou em intervalo de tempo predeterminado.
- e.g., múltiplos processos competindo por recursos e por tempo de processador, cria condições para o atendimento de todas as restrições e, por isso, nenhum sistema é completamente determinístico.
- .. razão com que o sistema operacional satisfaça requisições dos processos em um sistema de tempo real depende da velocidade com que responde às interrupções e eventos.
- ... bem como pela capacidade de tratar as requisições dentro de um intervalo de tempo que satisfaça os “deadlines” impostos pelas requisições dos processos de tempo real.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- ... uma medida útil da habilidade do sist. oper. funcionar deterministicamente é o atraso máximo desde a ocorrência de uma interrupção de alta prioridade até o seu reconhecimento.
- ... em sistemas não de tempo real, este atraso é $\geq 10s$ (dezenas) e $\leq 100s$ (centenas) de mili segundos (ms).
- ... já em sistemas de tempo real este atraso está entre alguns poucos microsegundos (10^{-6} seg.) a um milissegundo (10^{-3} seg.)

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- **“responsiveness” = “capacidade de resposta”** .. esta relacionada com quão rápido o sistema operacional atende uma requisição / interrupção após reconhecer a sua ocorrência;
- ... quão menor o tempo para responder a uma interrupção, maior será a capacidade de resposta do sistema.
- Capacidade de resposta envolve questões como:
- 01 - parcela de tempo necessária para iniciar o tratamento da interrupção, ou seja, iniciar “Interrupt Service Routine” (ISR).
- 02 - parcela de tempo necessária para executar a rotina de tratamento da interrupção, normalmente dependente da plataforma de hardware.
- 03 - efeito de loops de interrupção, situações em que uma rotina ISR pode ser interrompida por uma outra interrupção, gerando atraso.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**determinism**” e “**responsiveness**” .. juntos podem reduzir o tempo de resposta para eventos externos ou interrupções.
- “**tempo de resposta**” - ... sistemas de tempo real precisam atender as restrições de tempo impostas por indivíduos, dispositivos e fluxos de dados externos aos sistema computacional e operacional.
- .. variável extremamente crítica em sistemas de tempo real.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- **“user control”** - ... essencial para que o usuário possa controlar com certo grau de refinamento a prioridade da tarefa.
- ... em sistemas operacionais não de tempo real, o usuário não tem controle sobre as funções de escalonamento ou no máximo pode prover orientações gerais.
- ... usuário deve ser capaz de distinguir entre “hard tasks” e “soft tasks” bem como especificar proporções relativas entre as classes de tarefas (processos e threads).
- ... cabe ao usuário especificar características como o uso de paginação; permutação de processos; processos a serem mantidos em memória; algoritmos de transferência de informações do disco; etc.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**reliability**” - habilidade do sistema realizar e manter suas funções em circunstâncias rotineiras bem como em circunstâncias hostis ou inesperadas, ou seja, continuamente.
- e.g., falha de processador em um sistema SMP não de tempo real pode resultar na redução do nível de serviço até que o processador com defeito seja recuperado ou substituído.
- ... já em sistemas de tempo real que respondem e controlam eventos em tempo real, a perda ou degradação de desempenho pode trazer consequências catastróficas.
- ... desde perda financeira até danos transitórios ou não em equipamentos ou até mesmo perda da vida.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**fail-soft operation**” .. característica que reflete a habilidade do sistema falhar de tal maneira que preserve tanto quanto possível sua capacidade dados, ou seja, se mantenha operacional.
- e.g., UNIX tradicional ao detectar corrupção de dado no kernel, encaminha mensagem de falha para o console do sistema;
- ... na sequência despeja o conteúdo da memória para o disco para mais tarde analisar a falha e termina a execução do sistema.
- ... em sistemas de tempo real tentar-se-á corrigir o problema ou minimizar os efeitos enquanto continua a ser executado.
- ... aspecto também importante é a “**stability**”, ou seja, nos casos onde for impossível atender todos os “**deadlines**”, o sistema irá atender os “deadlines” das tarefas com maior prioridade e criticidade.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- Para atender os requisitos anteriores, sistemas de tempo real tipicamente contemplam as seguintes características:
- .. rápido chaveamento de processos e threads.
- .. menor tamanho incluindo as funcionalidades mínimas.
- .. habilidade para responder a eventos externos rapidamente.
- .. comunicação entre processos por semáforos, sinais ou eventos.
- .. uso de arquivos sequenciais para acúmulo rápido de dados.
- .. escalonamento preemptivo baseado em prioridade.
- .. redução de intervalos nos quais interrupções estão desabilitadas.
- .. primitivas para atrasar/recuperar tarefas por tempo fixo.
- .. alarmes especiais e timeouts.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

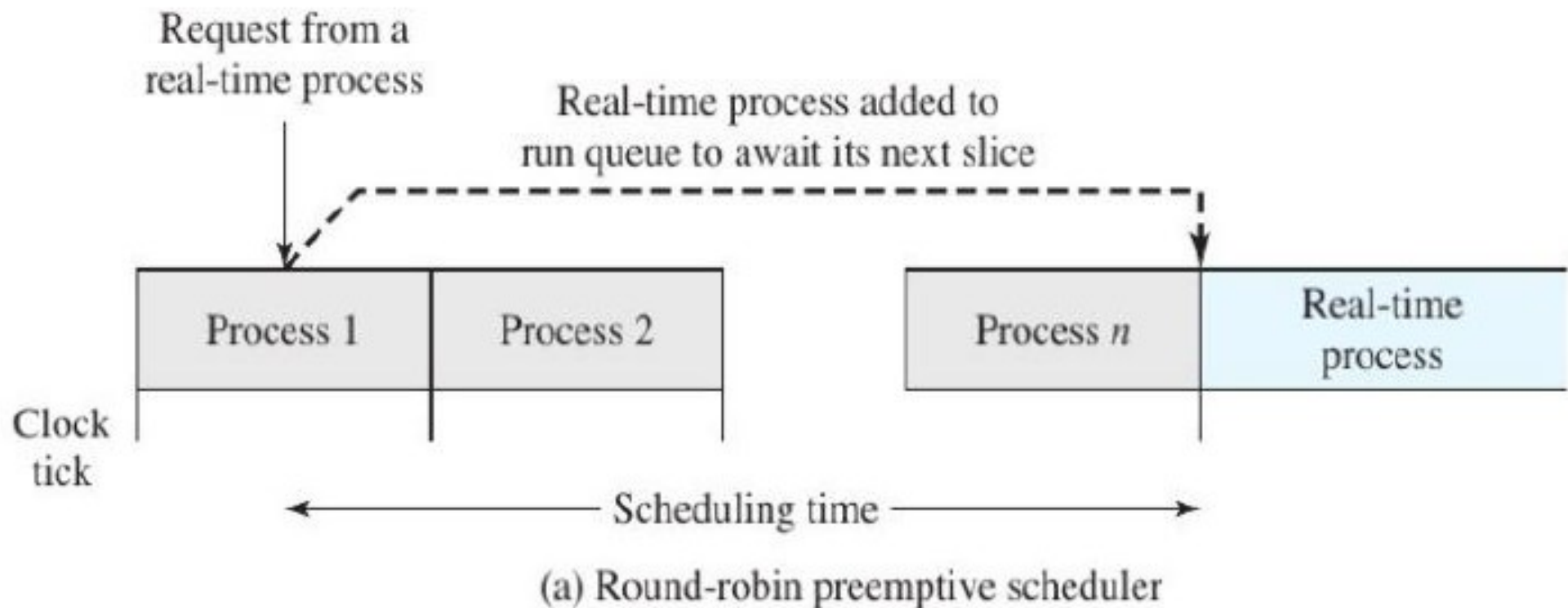
... 10.2.2 – Characteristics of Real-Time OS

- Aplicações de Tempo Real tipicamente requerem tempo de resposta determinístico dentro de uma faixa estreita, p.ex., 10 a 100 micro segundos (μs), e sob um amplo conjunto de condições.
- e.g., ... em um escalonador preemptivo que utiliza o round robin, uma tarefa de tempo real pode ser adicionada a fila de pronto para esperar pelo próximo “time slice”.
- ... vejamos alguns exemplos dos problemas presentes no escalonamento de aplicações de tempo real.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

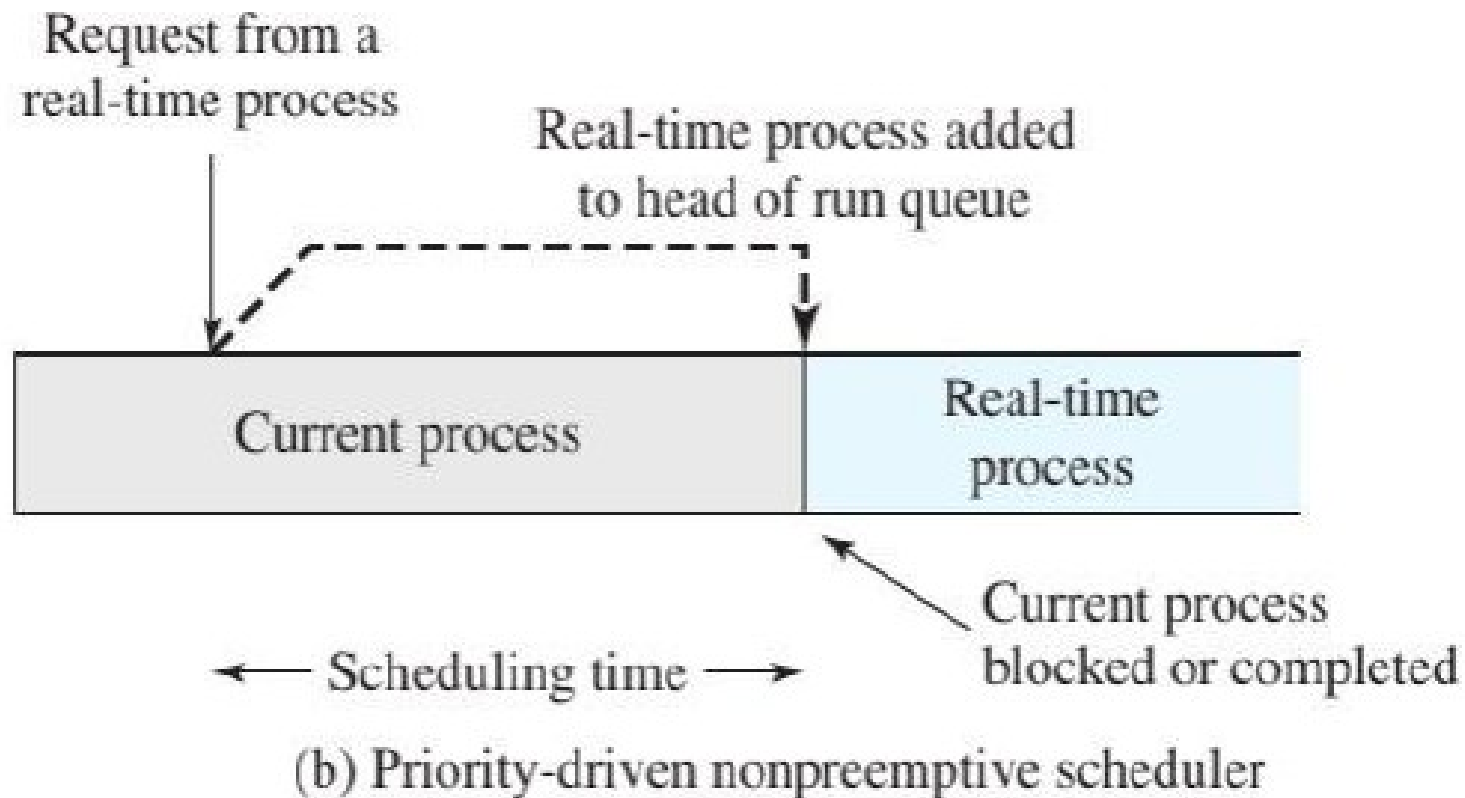
- ... neste caso, o tempo de escalonamento pode não atender as restrições de tempo impostas pela aplicação.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

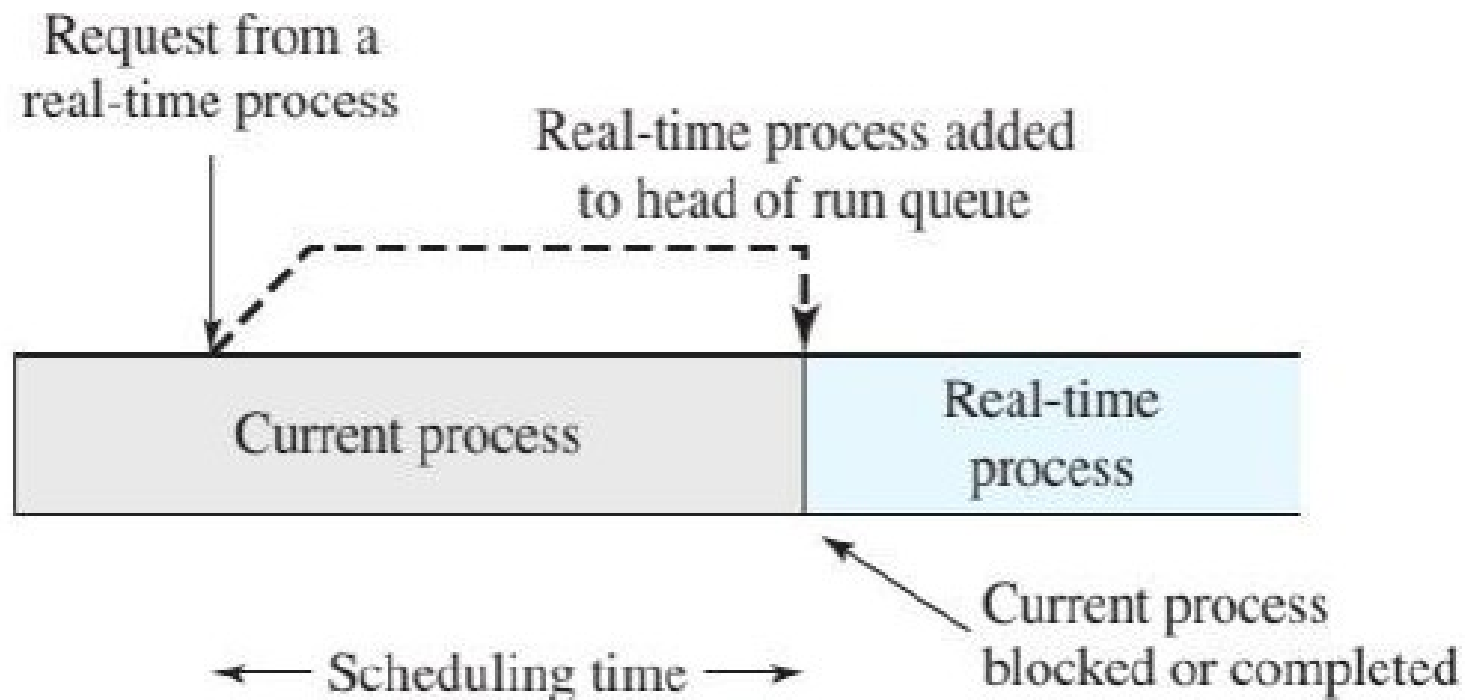
- ... em um escalonador não preemptivo, pode-se usar um mecanismo de escalonamento com prioridade para conceder às tarefas de tempo real uma maior prioridade.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- ... tarefa de tempo real que está pronta pode ser escalonada tão logo o processo corrente bloqueie ou complete.

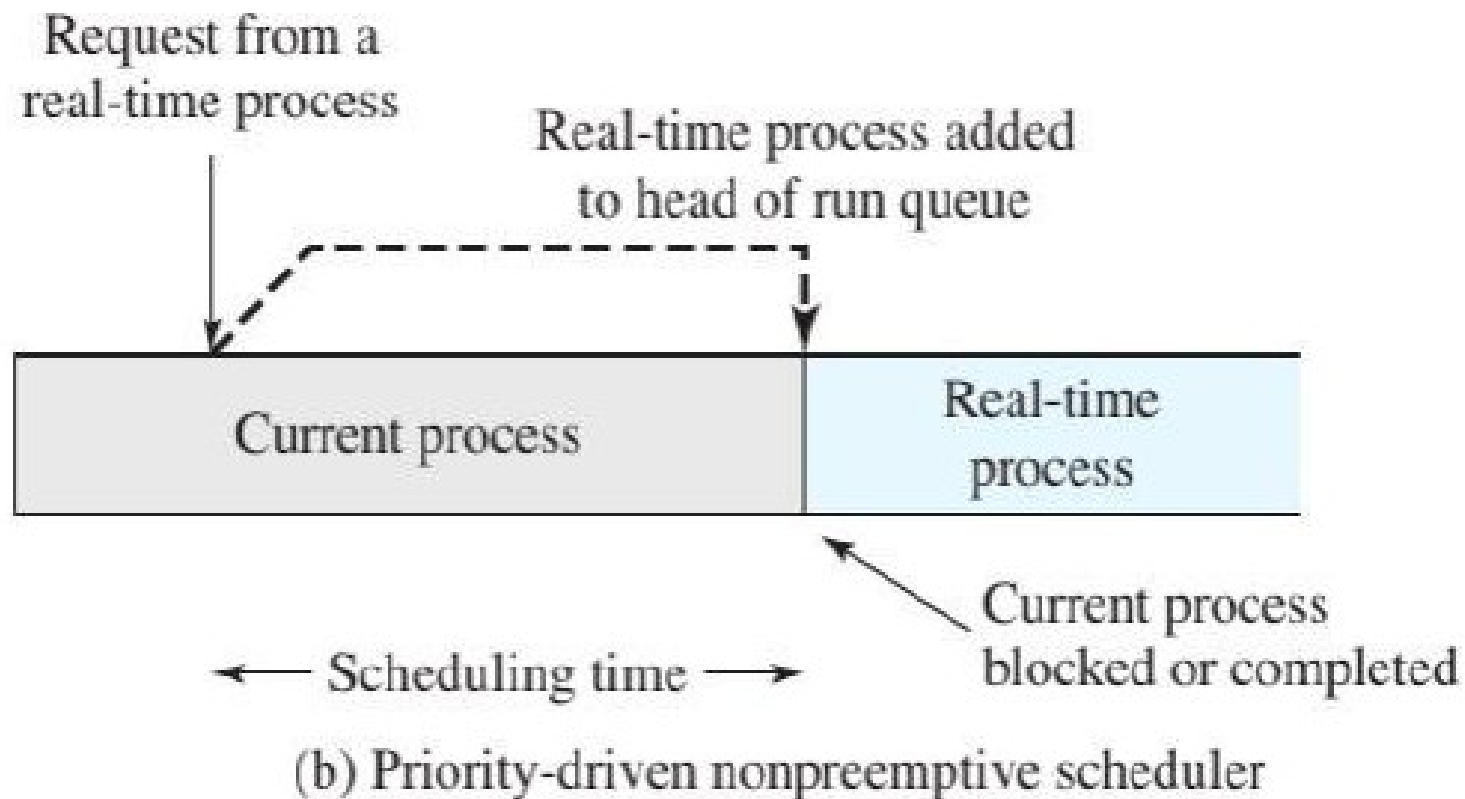


(b) Priority-driven nonpreemptive scheduler

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

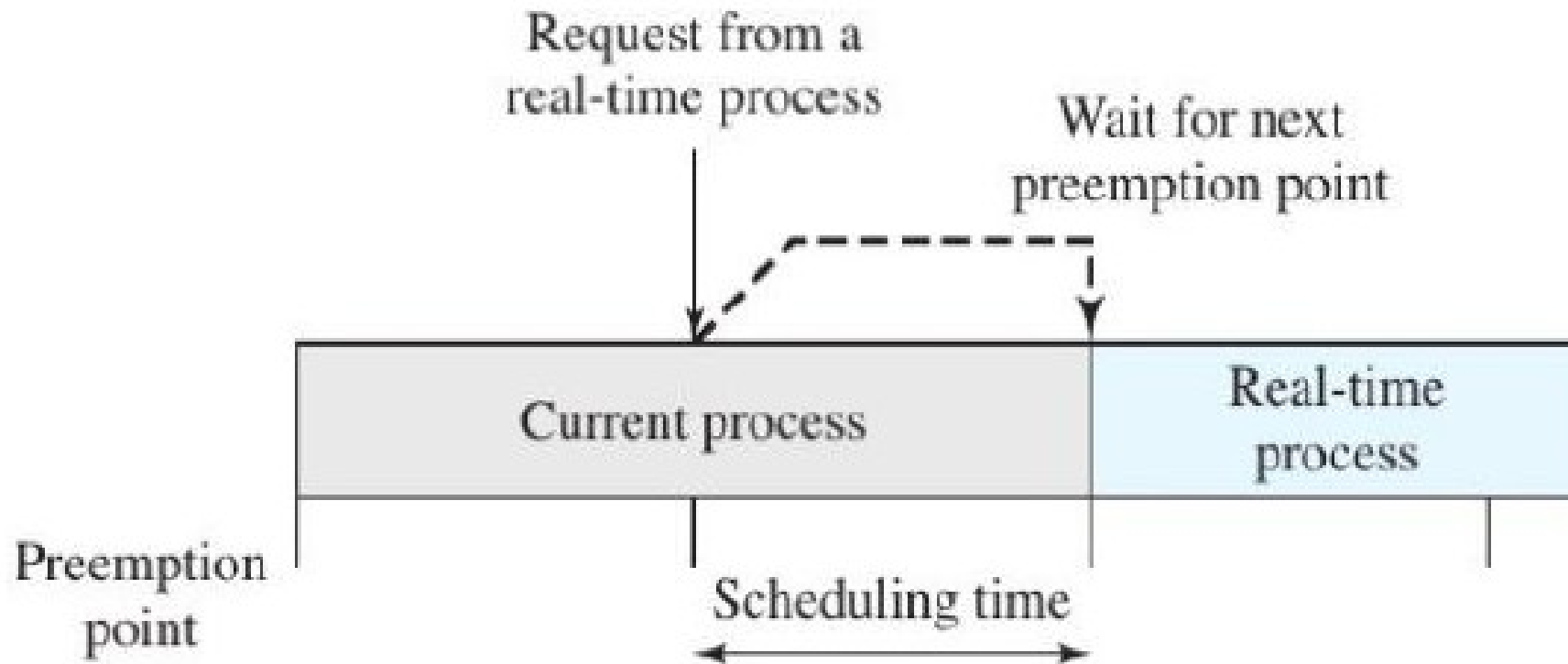
- ... isto pode levar a um atraso de alguns segundos se uma tarefa lenta de baixa prioridade estivesse executando no momento crítico » abordagem não aceitável.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**abordagem promissora**” .. combinar prioridades e interrupções tendo por base o relógio do sistema computacional.

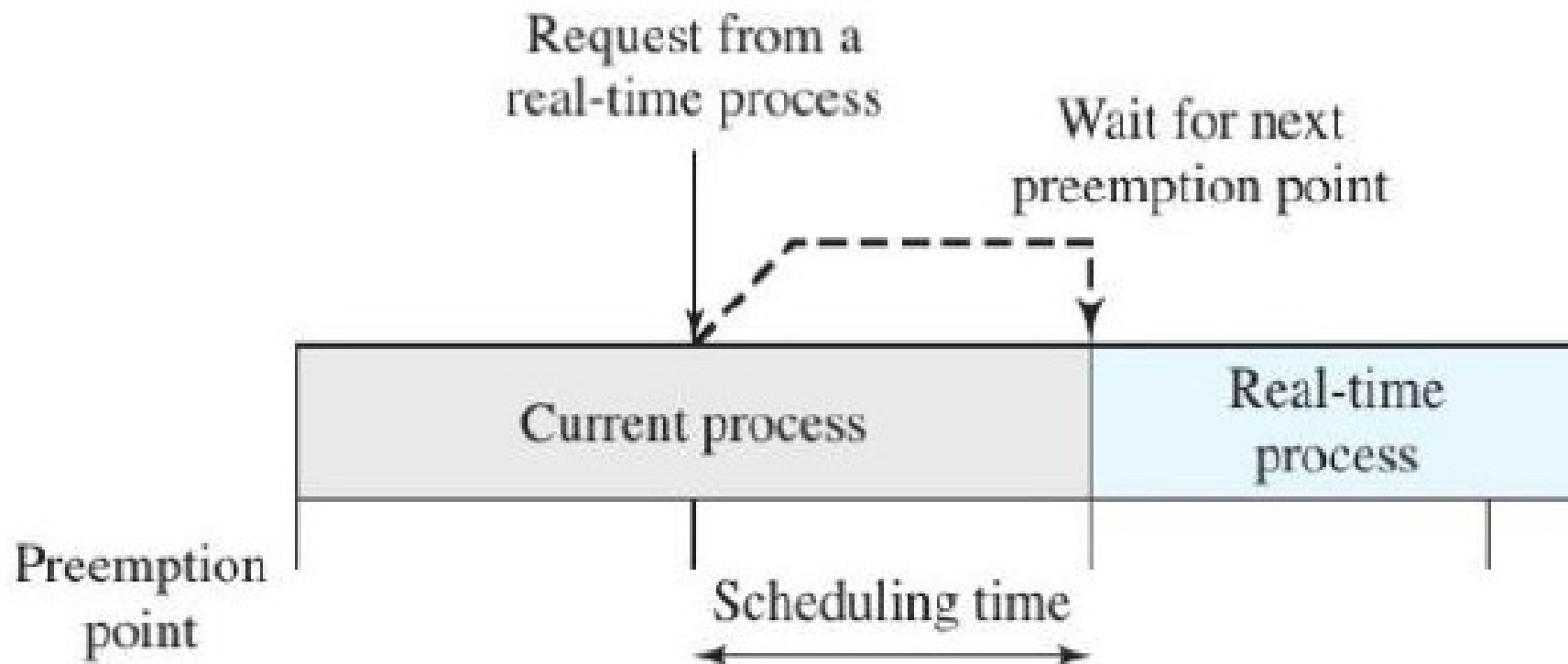


(c) Priority-driven preemptive scheduler on preemption points

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- ... assim, pontos de preempção em intervalos regulares, permitem que uma tarefa que está executando seja preemptada em favor de uma tarefa de maior prioridade.

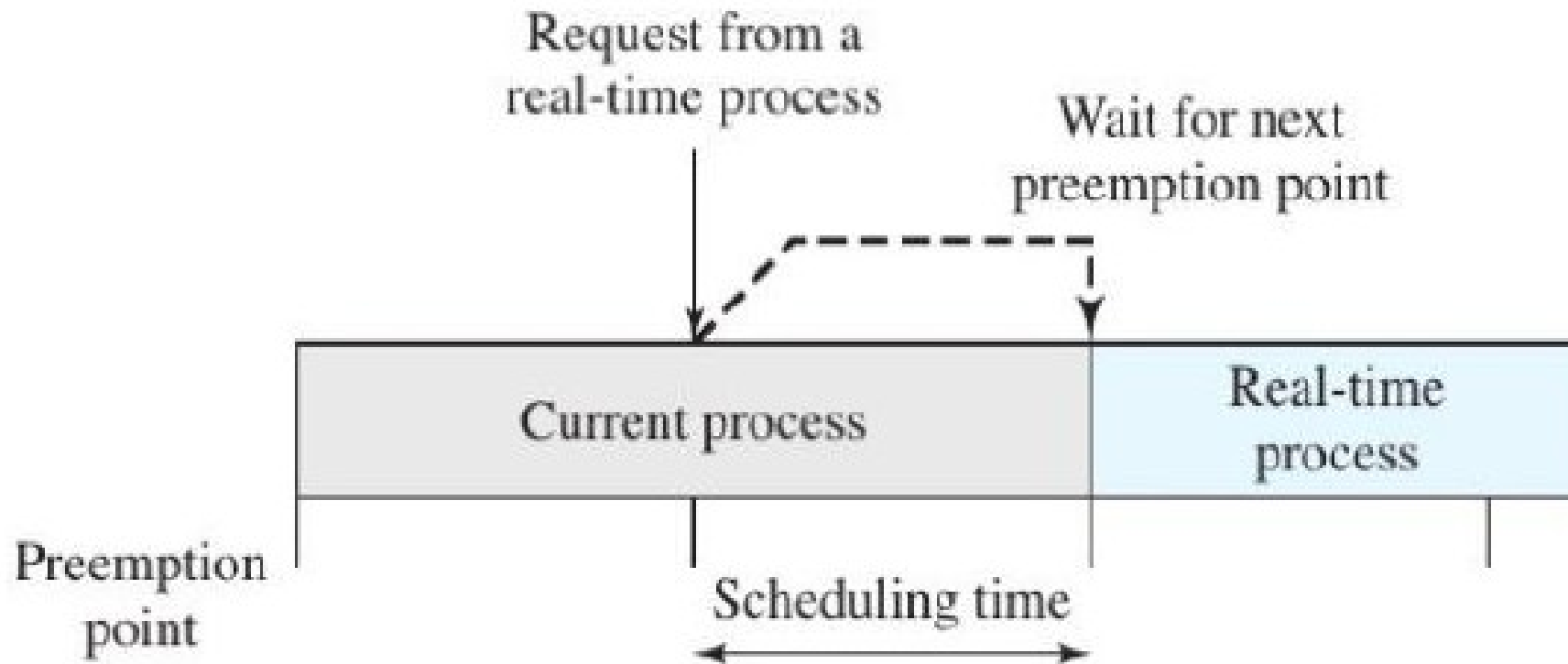


(c) Priority-driven preemptive scheduler on preemption points

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**conclusão**” .. embora esta abordagem seja adequada para algumas aplicações de tempo real, esta abordagem não é suficiente para as demais aplicações de tempo real.

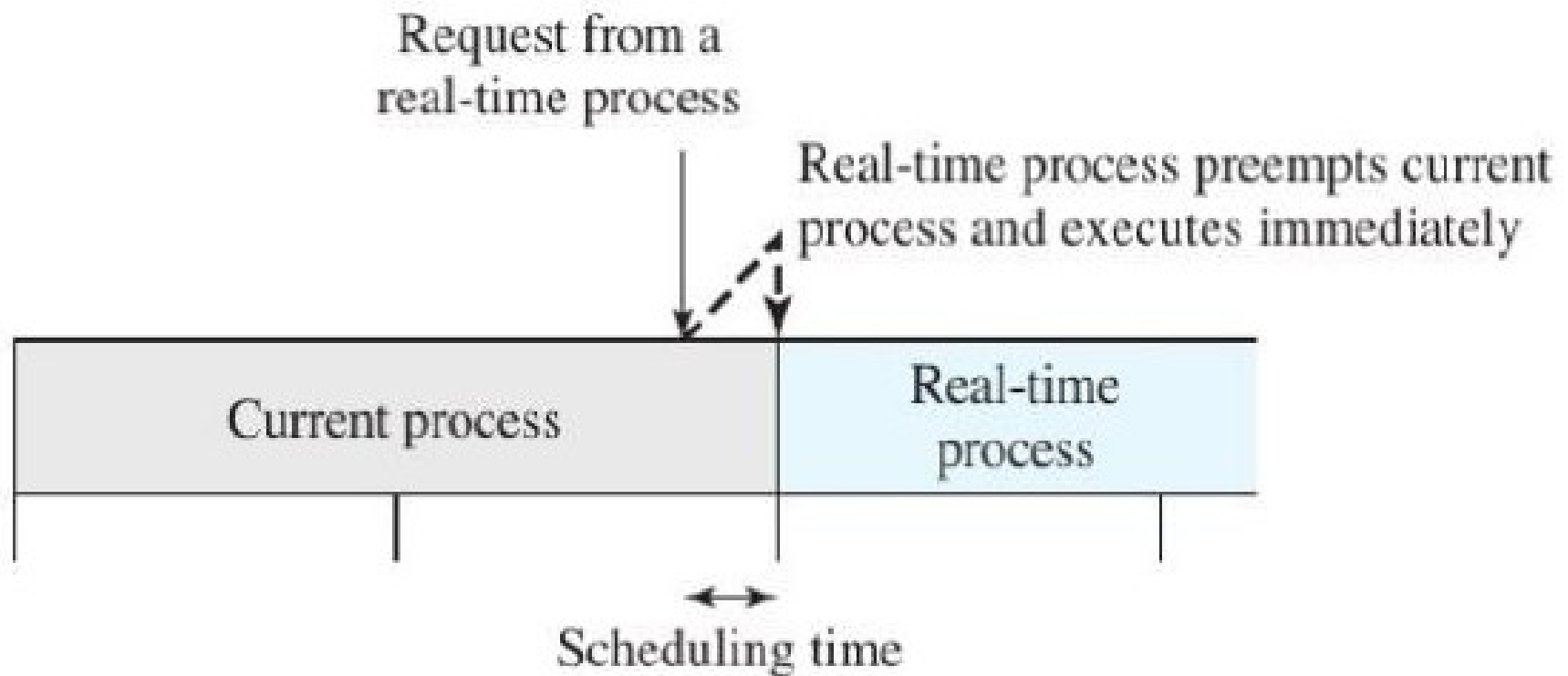


(c) Priority-driven preemptive scheduler on preemption points

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

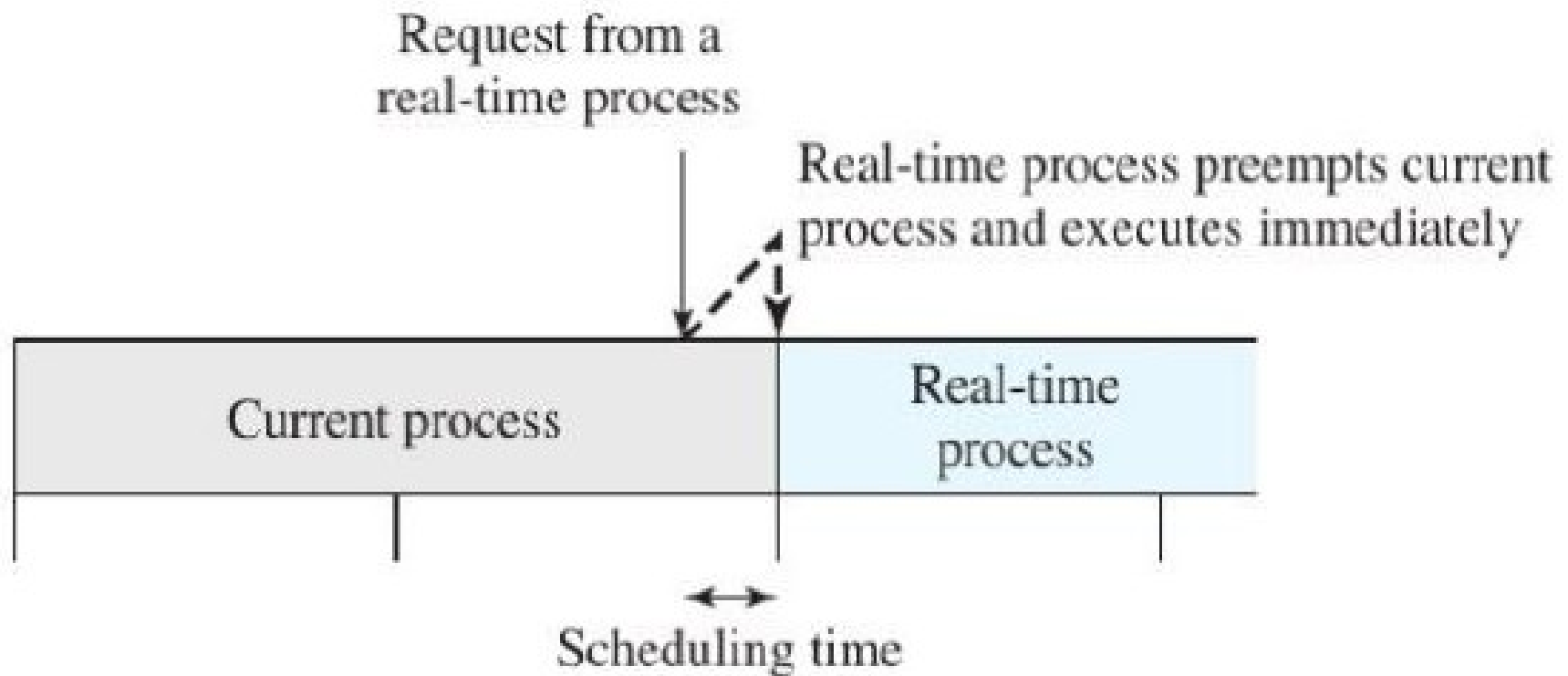
- .. nas demais aplicações de tempo real, o sistema operacional responde a uma interrupção imediatamente, a menos que o sistema esteja sob uma seção de bloqueio de código crítico.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.2 – Characteristics of Real-Time OS

- “**immediate preemption**” .. sistema operacional responde a uma requisição (interrupção) de tempo real imediatamente.



10.2.3 – Real-Time Scheduling

- “**real-time scheduling**” .. como observado por RAMA 1994, as várias abordagens de escalonamento dependem de:
 - .. quando o sistema realizou análise de escalonabilidade;
 - .. análise de escalonabilidade é estática ou dinâmica;
 - .. resultado da análise produziu um escalonamento de acordo com as tarefas que são solicitadas em tempo de execução.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.3 – Real-Time Scheduling

- Baseado em tais considerações, 04 classes são propostas:
- **“static table-driven approaches”** - contempla uma análise estática dos escalonamentos viáveis, ou seja, o escalonador determina em tempo de execução quando a tarefa deve ser executada.
- **“static priority-driven preemptive approaches”** - contempla a análise estática para atribuir prioridades às tarefas, para na sequência um escalonador preemptivo tradicional ser usado.
- **“dynamic planning-based approaches”** - uma tarefa é aceita para ser executada somente se suas restrições de tempo são satisfeitas, ou seja, a viabilidade é determinada dinamicamente.
- **“dynamic best effort approaches”** - não contempla análise de viabilidade, assim o sistema procura atender todos os “deadlines” e pode suspender algum processo já iniciado cujo “deadline” não foi satisfeito.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

10.2.4 – Deadline Scheduling

- **“aplicações de tempo real”** .. tem como principal componente a garantia de que tarefas serão iniciadas ou finalizadas em instantes de tempo previamente estabelecidos.
- ... independente do conflito e demanda por recursos, carga de processamento, falhas de software ou hardware.
- **“escalonamento de tempo real”** .. baseadas na obtenção de informações adicionais para cada tarefa do conjunto de tarefas:
- ... **“ready time”**; **“starting deadline”**; **“completion deadline”**;
- ... **“processing time”**; **“priority”**, etc.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- Dentre as inúmeras propostas para escalonamento de tempo real, todas estão baseadas na obtenção de informações adicionais para cada tarefa dentro do conjunto de tarefas:
- “**ready time**” .. tempo no qual a tarefa se encontra pronta para ser executada e, para o caso de tarefas periódicas, constitui-se na sequência de marcas de tempos que deverá ser executada.
- “**starting deadline**” .. tempo no qual a tarefa deve se iniciada.
- “**completion deadline**” .. tempo no qual a tarefa deve se completada e, normalmente, apenas um dos tempos - “starting” ou “completion” são utilizados, raramente ou nunca os dois tempos.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- (cont.) ... dentre as inúmeras propostas para escalonamento de tempo real, todas estão baseadas na obtenção de informações adicionais para cada tarefa do conjunto de tarefas:
- “**processing time**” .. tempo necessário para executar a tarefa, sendo que em alguns casos é fornecido e em outros o sistema calcula a média exponencial das últimas execuções.
- “**resource requirements**” .. conjunto de recursos além do processador exigidos pela tarefa enquanto em execução (ciclo de vida).

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

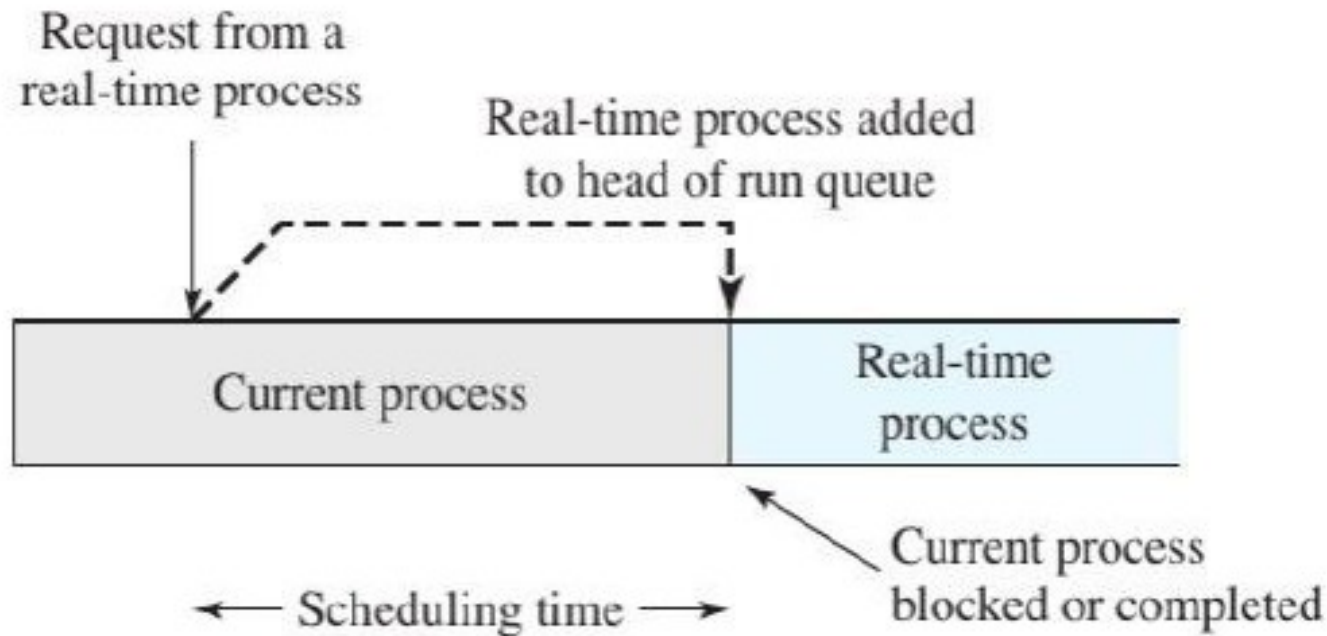
... 10.2.4 – Deadline Scheduling

- (cont.) ... dentre as inúmeras propostas para escalonamento de tempo real, todas estão baseadas na obtenção de informações adicionais para cada tarefa do conjunto de tarefas:
- “**priority**” .. mede a importância relativa da tarefa, sendo que tarefas “hard real-time” tem prioridade absoluta, ou seja, o sistema falha se o “deadline” não for satisfeito;
- ... se o sistema deve continuar executando seja qual for o problema, então ambas tarefas “hard real-time” e “soft real-time” terão prioridades relativas para guiar o escalonador;
- “**subtask structure**” .. tarefa pode ser decomposta em sub- tarefas mandatórias e sub-tarefas opcionais, sendo que as sub-tarefas mandatórias serão tratadas como “hard real-time”.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- “**critical design issue**” .. questão crítica de projeto é a preempção, p.ex., quando o “starting deadline” é especificado, faz sentido o escalonador ser o escalonador não preemptivo.
- .. pois é responsabilidade da tarefa se bloquear após completar a porção crítica ou mandatória de sua execução.

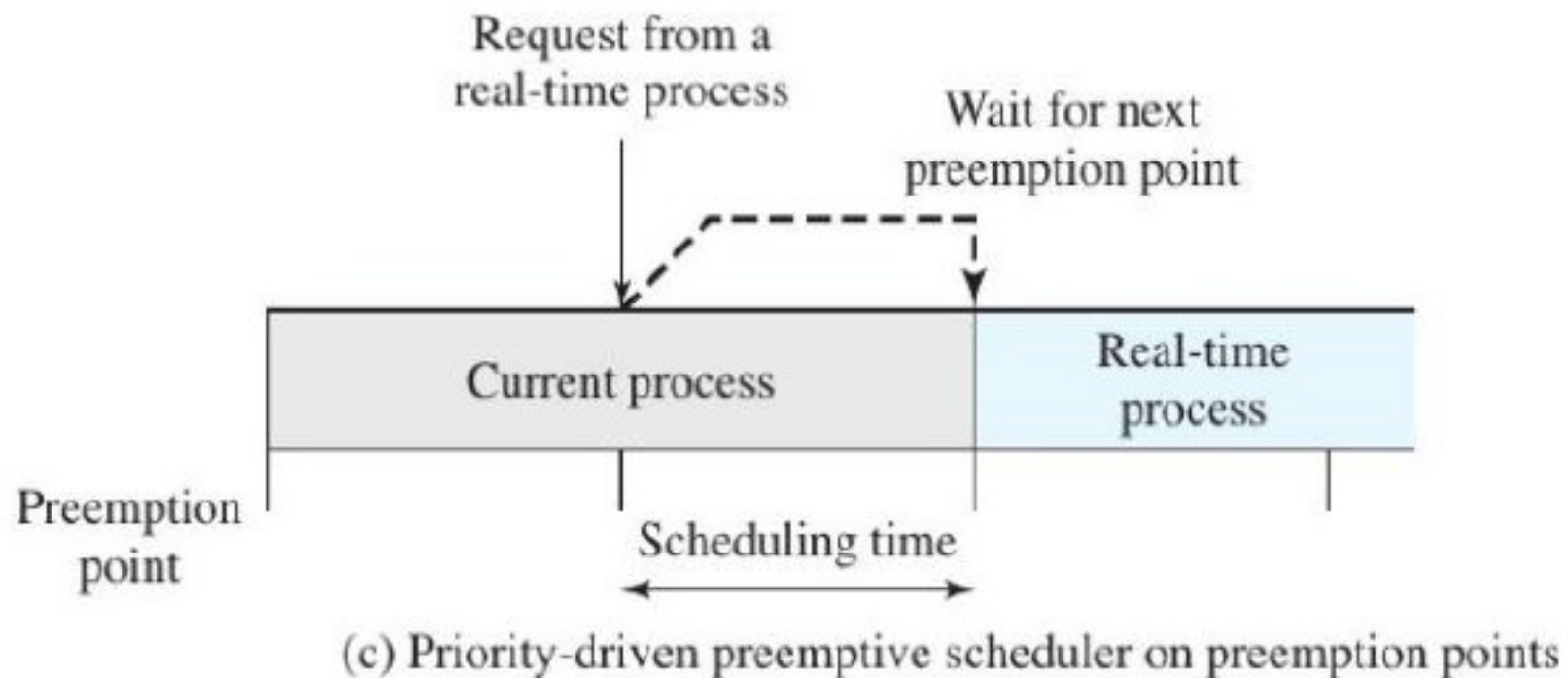


(b) Priority-driven nonpreemptive scheduler

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

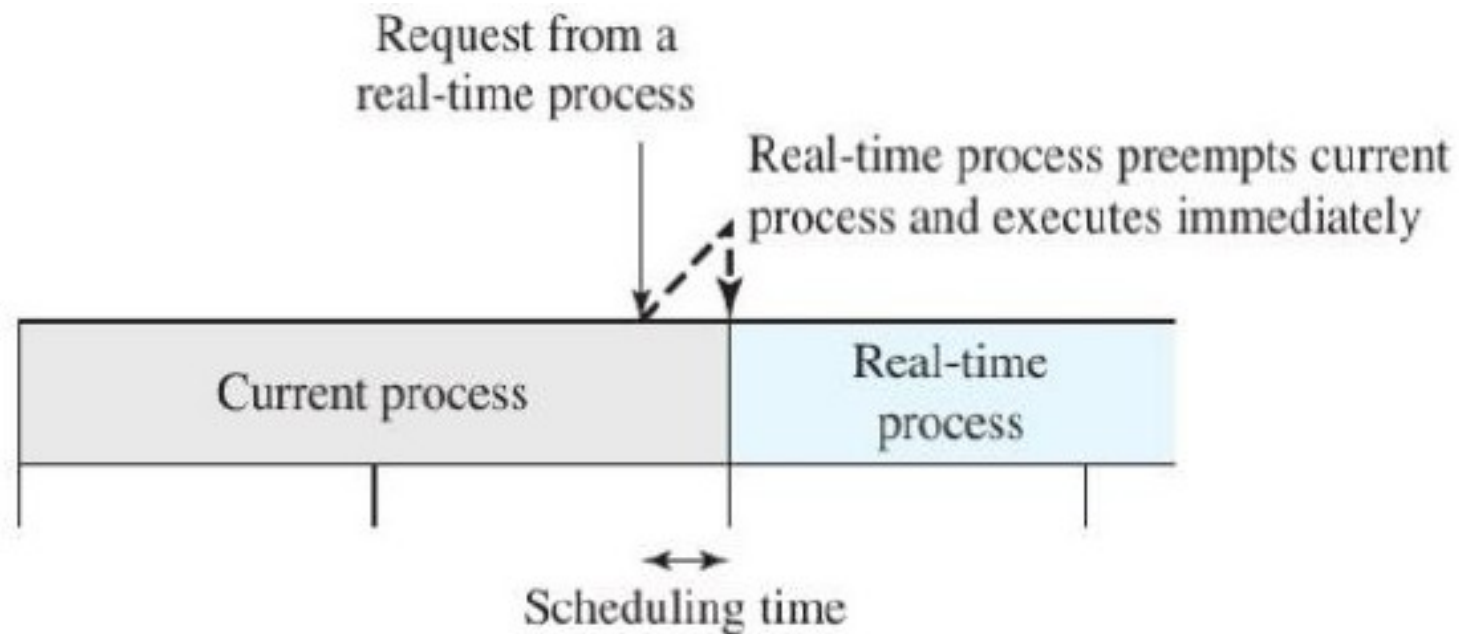
- ... para um sistema com “completion deadline”, a estratégia preemptiva é mais apropriada como pode ser constatado abaixo.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- e.g., ... se a tarefa “X” está executando e a tarefa “Y” está pronta, pode existir o contexto no qual o caminho para ambos “X” e “Y” satisfazerem seus “deadlines” seja preemptar “X”, executar “Y” até o seu “deadline” ser satisfeito e recuperar “X” para completar.



(d) Immediate preemptive scheduler

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- e.g., ... como exemplo de tarefas periódicas com “completion deadline”, considere um sistema que coleta e processa dados de um conjunto de 02 sensores, “A” e “B”.
- ... “deadline” para coletar dados do sensor “A” se dá a cada 20 ms e o “deadline” para coletar dados do sensor “B” a cada 50 ms;
- ... para processar cada amostra, incluindo o “overhead” do sist. oper. são necessários 10 ms para cada amostra de “A” e 25 ms para processar cada amostra do sensor “B”.
- ... tabela a seguir sintetiza o perfil de execução das 02 tarefas e as figuras na sequência comparam 03 técnicas de escalonamento utilizando o perfil apresentado na tabela.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

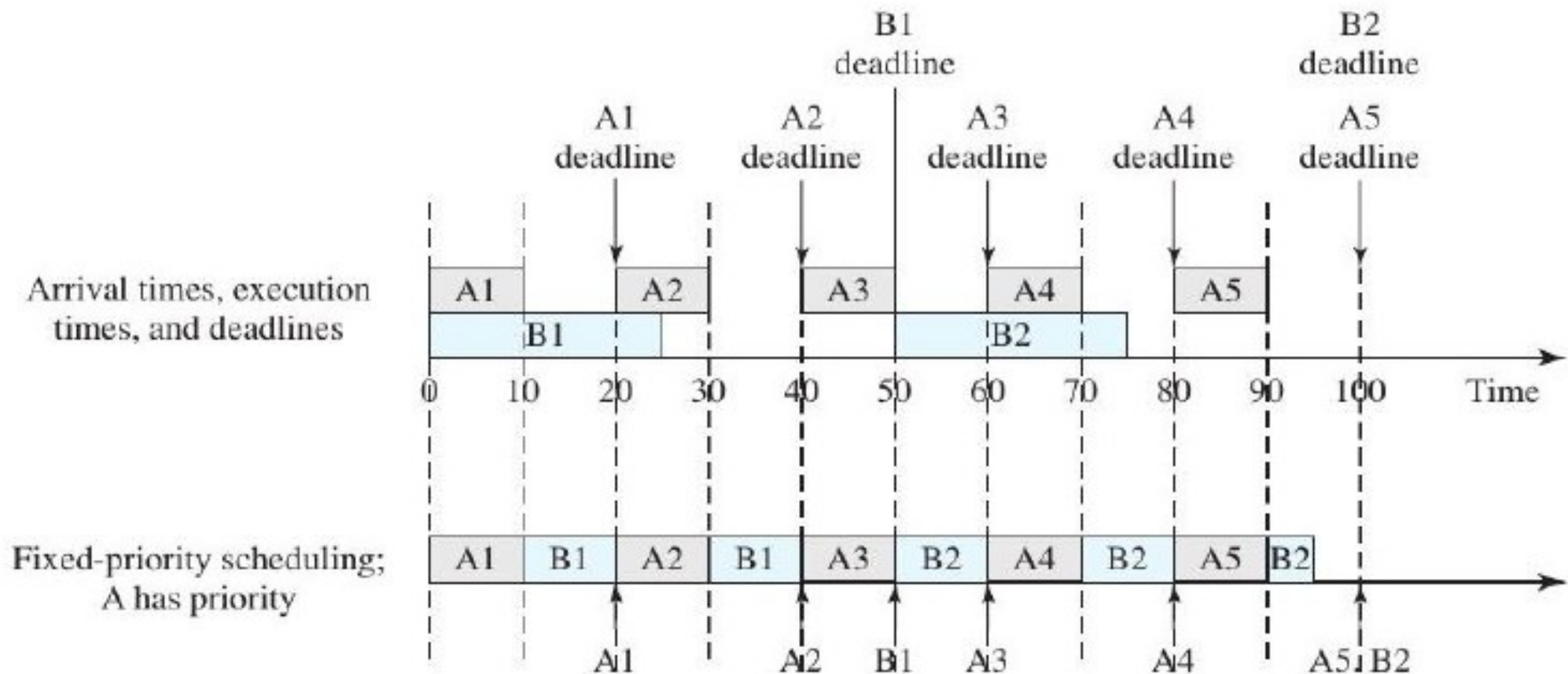
- Tab. 10.2 - Síntese do perfil de execução das 02 tarefas “A” e “B” que coletam e processam dados de 02 sensores, “A” e “B”.

Process	Arrival Time	Execution Time	Ending Deadline
A(1)	0	10	20
A(2)	20	10	40
A(3)	40	10	60
A(4)	60	10	80
A(5)	80	10	100
•	•	•	•
•	•	•	•
•	•	•	•
B(1)	0	25	50
B(2)	50	25	100
•	•	•	•
•	•	•	•
•	•	•	•

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

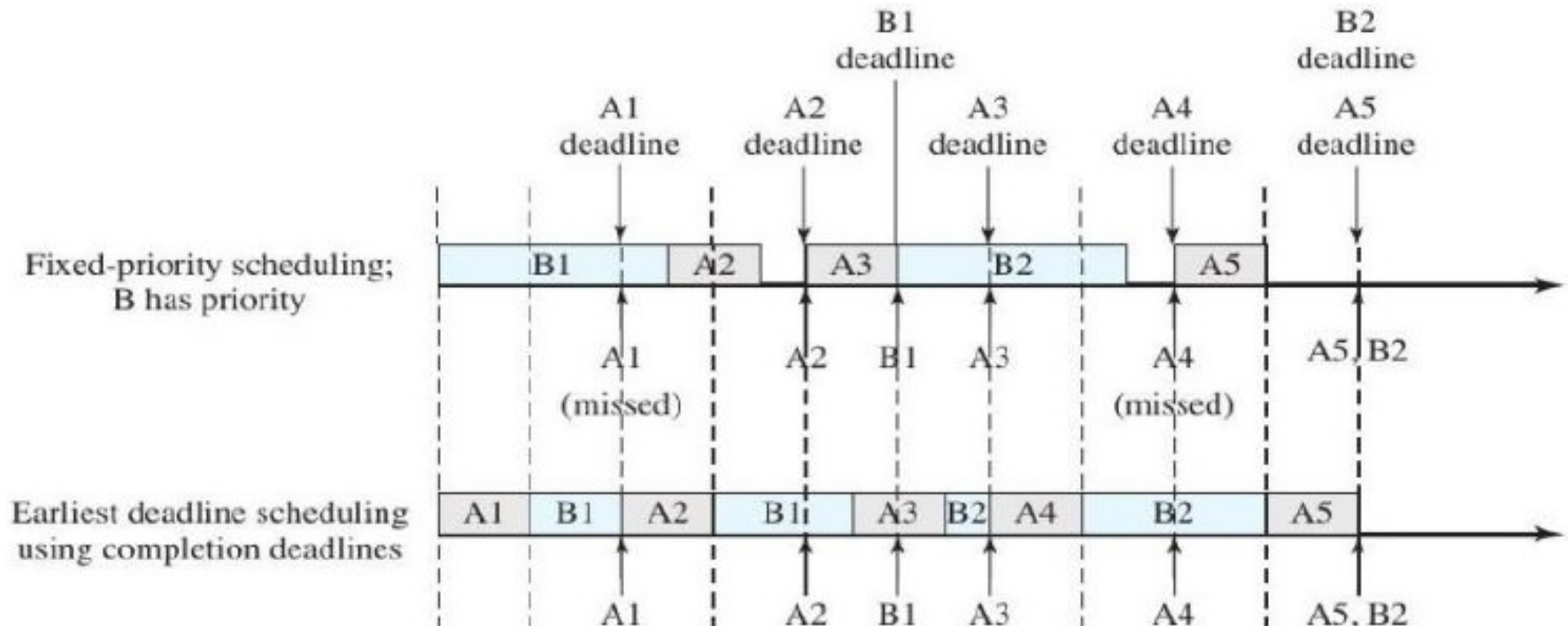
- Escalonamento de Tarefas Periódicas de Tempo Real com “completion deadline” baseado na Tab. 10.2



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- Escalonamento de Tarefas Periódicas de Tempo Real com “completion deadline” baseado na Tab. 10.2

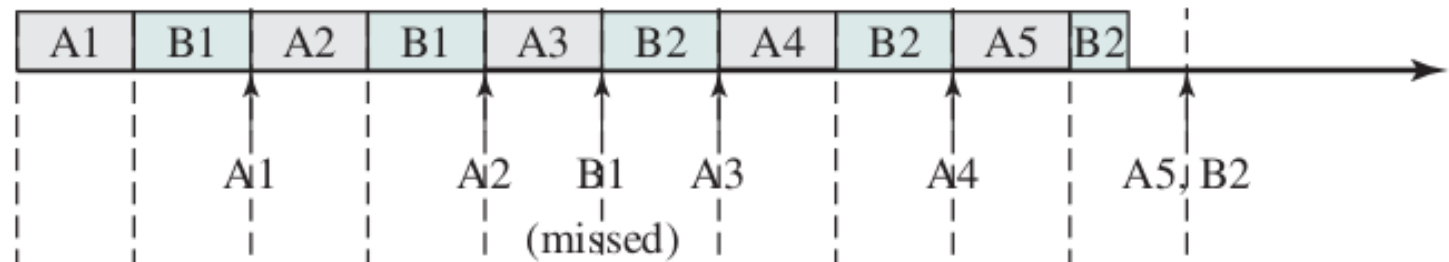


10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

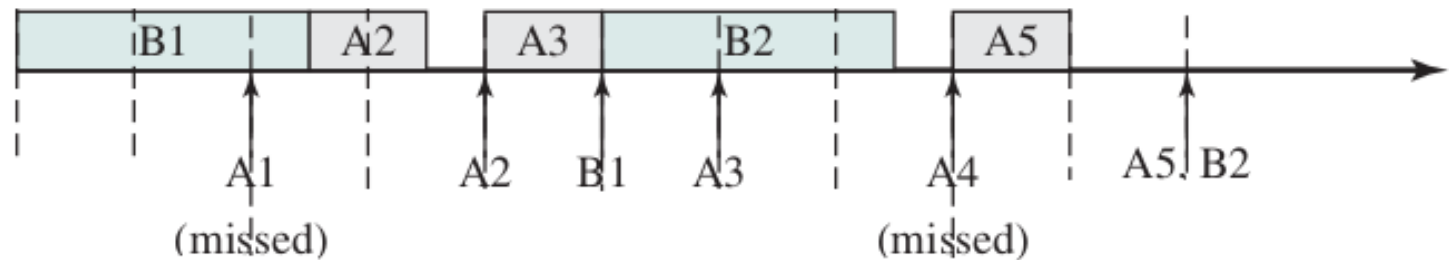
... 10.2.4 – Deadline Scheduling

- .. se A tiver maior prioridade, a primeira instância de B terá apenas 20 ms de processamento, ou seja, “deadline” não será satisfeito.
- .. se B tiver maior prioridade, então A irá perder o 1o “deadline”.

Fixed-priority scheduling;
A has priority



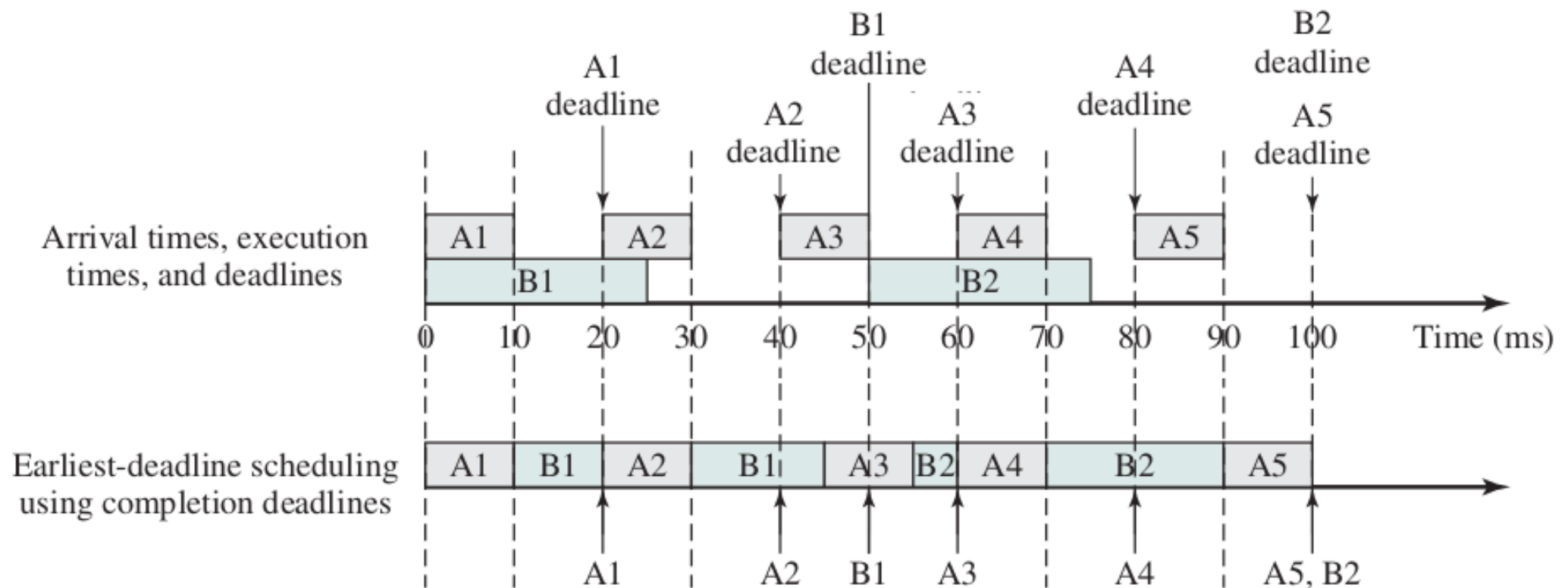
Fixed-priority scheduling;
B has priority



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

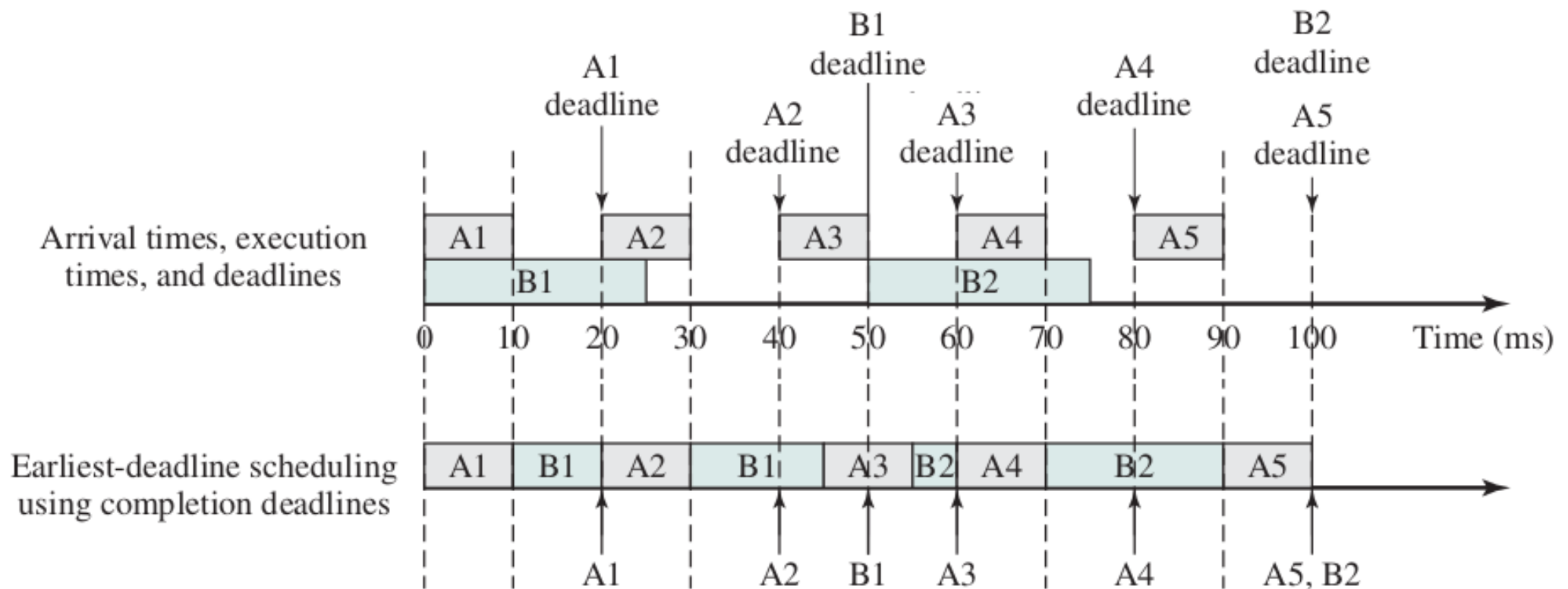
- Diagrama final mostra o uso do escalonamento pelo “deadline” que completa primeiro, ou seja, em $t = 0$ A1 e A2 chegam, mas como o “deadline” de A1 completa primeiro ele é escalonado.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

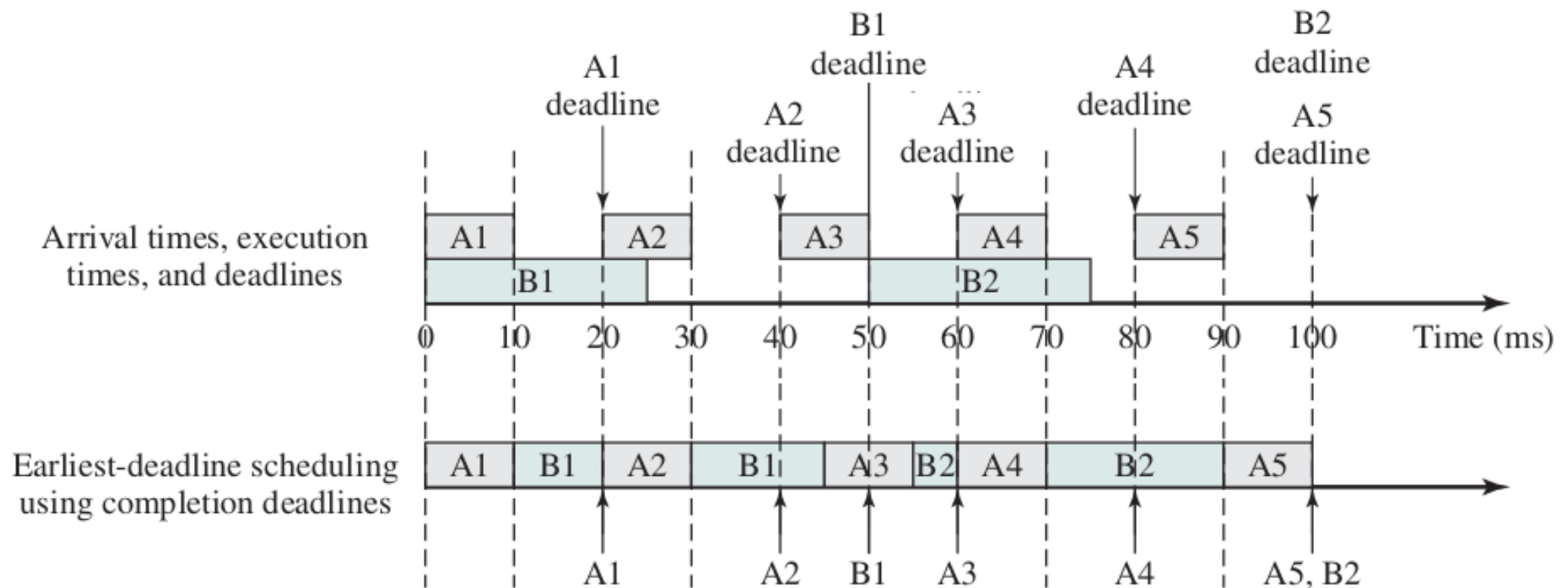
- ... ao completar, o processador é repassado para B1 que executa por 10 ms e como em $t = 20$ A2 chega e tem o menor “deadline”, B1 é preemptado e reassume o processador em $t = 30$ ms.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.4 – Deadline Scheduling

- ... a partir deste ponto o escalonamento do processo com o “deadline” que acontece primeiro faz a diferença.



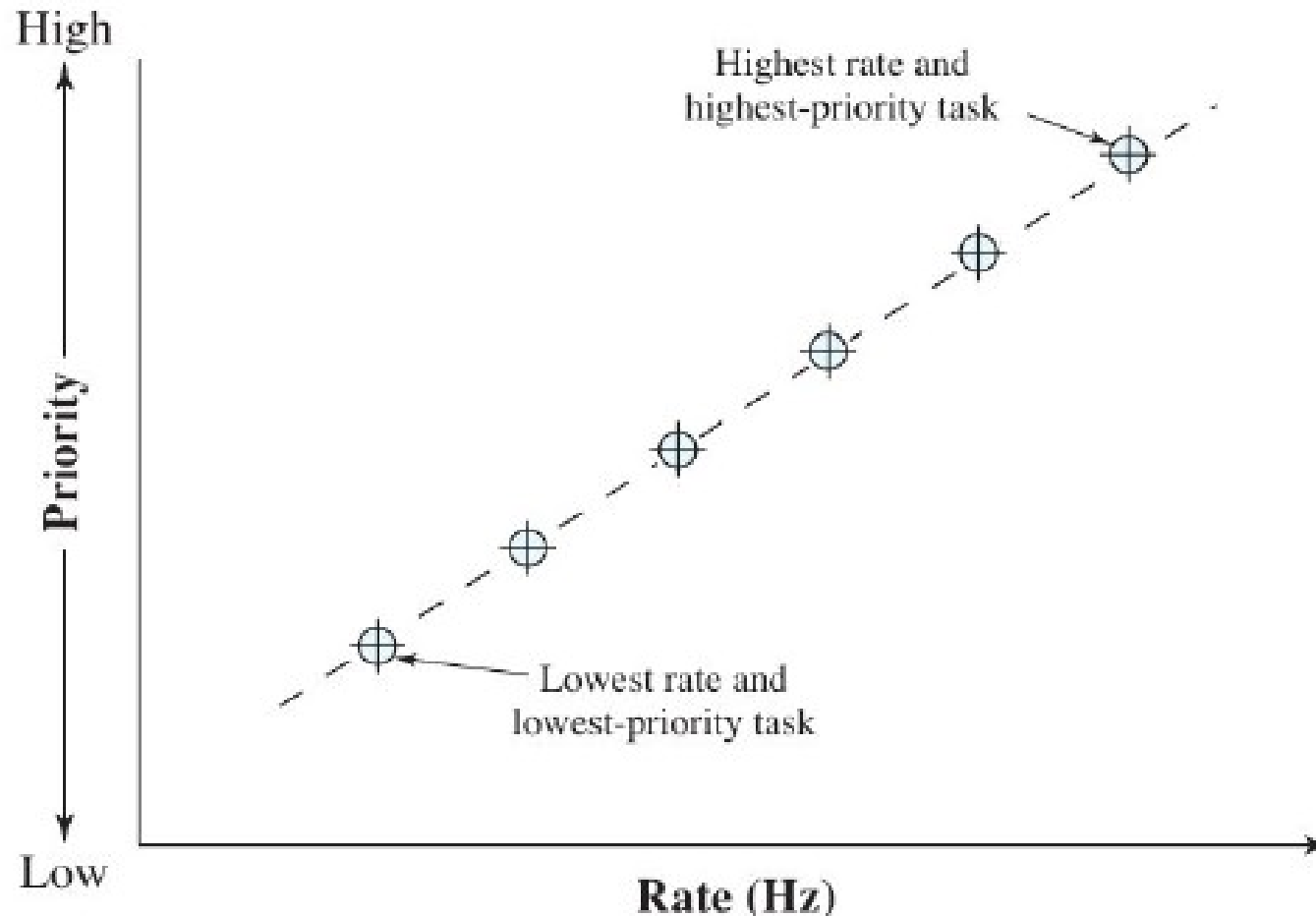
10.2.5 – Rate Monotonic Scheduling

- **“Rate Monotonic Scheduling”** (RMS) .. método para resolver conflitos de escalonamento para tarefas periódicas no qual a atribuição de prioridade para as tarefas tem por base os seus períodos.
- ... a tarefa com a maior prioridade é a tarefa com o menor período, a tarefa com a segunda maior prioridade é aquela com o segundo menor período e, assim, por diante;
- ... se plotarmos a prioridade de tarefas como uma função da sua taxa, o resultado é um função que cresce monotonicamente, daí o nome RMS – Rate Monotonic Scheduling.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.5 – Rate Monotonic Scheduling

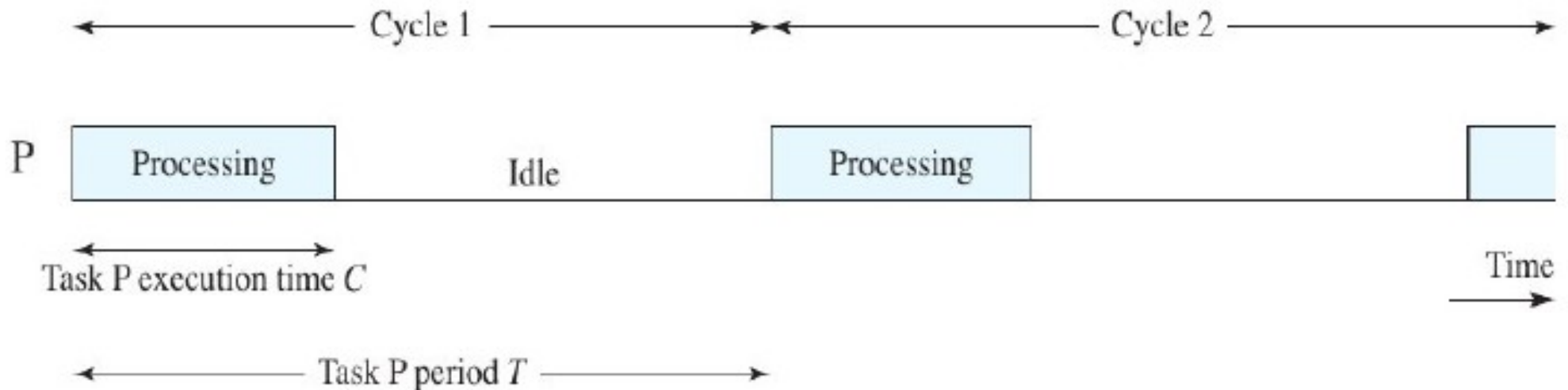
- ... se plotarmos a prioridade de tarefas como uma função da sua taxa, o resultado é um função que cresce monotonicamente, daí o nome RMS – Rate Monotonic Scheduling.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.5 – Rate Monotonic Scheduling

- .. figura ilustra os parâmetros relevantes para tarefas periódicas.
- e.g., T = período entre a chegada de duas instâncias consecutivas para uma mesma tarefa; F – frequência com a que a tarefa de período T se repete no tempo (Hz).



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.5 – Rate Monotonic Scheduling

- “**effectiveness**” .. uma medida de efetividade para o algoritmo de escalonamento periódico é a garantia ou não de que todos os “hard deadlines” serão satisfeitos.
- e.g., considere “n” tarefas com período fixo e tempo de execução, então para garantir que todos os “deadlines” sejam satisfeitos a seguinte inequação é válida: $C_1/T_1 + C_2/T_2 + \dots + C_n/T_n \leq 1$
- ... esta equação estabelece um limite sobre o nro de tarefas que podem com sucesso serem escalonadas, no entanto, o limite pode ser menor para algum algoritmo em particular, p.ex. RMS:
- $C_1/T_1 + C_2/T_2 + \dots + C_n/T_n \leq n * (2^{1/n} - 1)$

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.5 – Rate Monotonic Scheduling

- Tab. 10.4 dá alguns valores para este limite superior e a medida que o nro. de tarefas cresce o limite do escalonamento converge para $\ln 2 =$ (aproximadamente) 0.693.

n	$n(2^{1/n} - 1)$
1	1.0
2	0.828
3	0.779
4	0.756
5	0.743
6	0.734
•	•
•	•
•	•
∞	$\ln 2 \approx 0.693$

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.5 – Rate Monotonic Scheduling

- e.g., considere o caso de 03 tarefas onde $U_i = C_i / T_i$
- Tarefa P1 com $C_1 = 20$; $T_1 = 100$ e $U_1 = 0.2$;
- Tarefa P2 com $C_2 = 40$; $T_2 = 150$ e $U_2 = 0.267$;
- Tarefa P3 com $C_3 = 100$; $T_3 = 350$ e $U_3 = 0.286$
- ... total de utilização de processador destas tarefas é $0.2 + 0.267 + 0.286 = 0.753$, enquanto que o limite superior usando o RMS é:
- $C_1/T_1 + C_2/T_2 + C_3/T_3 \leq 3 * (21/3 - 1) = 0.779$
- ... porque o total de utilização do processador para as 03 tarefas é menor que o limite para o RMS, ou seja, $0.753 < 0.779$ (RMS), então se o RMS é usado, todas as tarefas serão escalonadas.

10.2.6 – Priority Inversion

- **“priority inversion”** .. fenômeno que ocorre em todo e qualquer escalonador baseado em prioridade, mas que é especialmente relevante em sistemas de tempo real.
- e.g., “rover robot” que aterrizou em Marte em 04/07/1997 para coletar e enviar dados à Terra, iniciou após alguns dias de missão reinicializações contínuas do seu sistema.
- ... após muito esforço da equipe do “Jet Propulsion Laboratory” (JPL), o problema foi rastreado como sendo “priority inversion”.
- **“priority scheduling”** . escalonamento baseado em prioridade, sistema irá sempre executar a tarefa com a maior prioridade.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- “**priority inversion**” .. ocorre quando circunstâncias internas ao sistema forçam a tarefa de maior prioridade esperar uma tarefa de menor prioridade se completar.
- e.g., ... isto acontece quando uma tarefa de baixa prioridade manteve um dado recurso objeto de compartilhamento e uma tarefa de maior prioridade solicitou acesso ao mesmo recurso.
- ... naturalmente que a tarefa de maior prioridade será bloqueada até que o recurso esteja disponível, ou seja, até que a tarefa de menor prioridade complete e libere o recurso.
- ... tarefa de maior prioridade recupera-se do bloqueio e, assim, há chance de que nenhuma restrição de tempo real seja violada.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- “**unbounded priority inversion**” .. condição mais séria que ocorre quando a duração da inversão de prioridade depende:
 - tempo exigido para tratar o recurso compartilhado;
 - ações não previsíveis de outras tarefas não relacionadas.
- e.g., bom exemplo para o este fenômeno é a “inversão de prioridade” experimentada pelo software da “Pathfinder” do “rover robot” era inversão de prioridade “c”
- Pathfinder incluiu 03 tarefas, em ordem decrescente de prioridade:
 - T1 – verifica periodicamente o sistema e software da espaçonave;
 - T2 – processa dados de imagens;
 - T3 – realizar testes ocasionais acerca do “status” dos equipamentos.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- ... após T1 executar, o temporizador é reinicializado para o seu valor máximo e, se o temporizador expirar, assume-se que o software está de alguma forma comprometido.
- ... neste caso, o processador é interrompido, todos os dispositivos são reinicializados e o software é completamente recarregado, na sequência testado e finalmente reinicializado.
- ... esta sequência de recuperação / reinicialização não se completa até o dia seguinte, portanto, é uma operação de alto custo.
- “**observação**” .. tarefas T#1 e T#3 compartilham estruturas de dados comuns, protegidas por um semáforo binário.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- Considerando que as tarefas T1 e T3 compartilham dados comuns e protegidos por um semáforo binário, seja a sequência eventos t_1 , t_2 , ... t_n que causou a inversão de prioridade ..
- t_1 .. T#3 inicia execução.
- t_2 .. T#3 bloqueia o semáforo “s” e entra na região crítica.
- t_3 .. T#1 cuja prioridade é maior do que T#3, preempta T3 e inicia a execução e em algum momento irá solicitar o mesmo recurso.
- t_4 .. T#1 tenta entrar na região crítica, mas é bloqueado pois o semáforo “s” já está bloqueado em T#3, então, T#3 recupera execução na sua região crítica (inversão de prioridade).

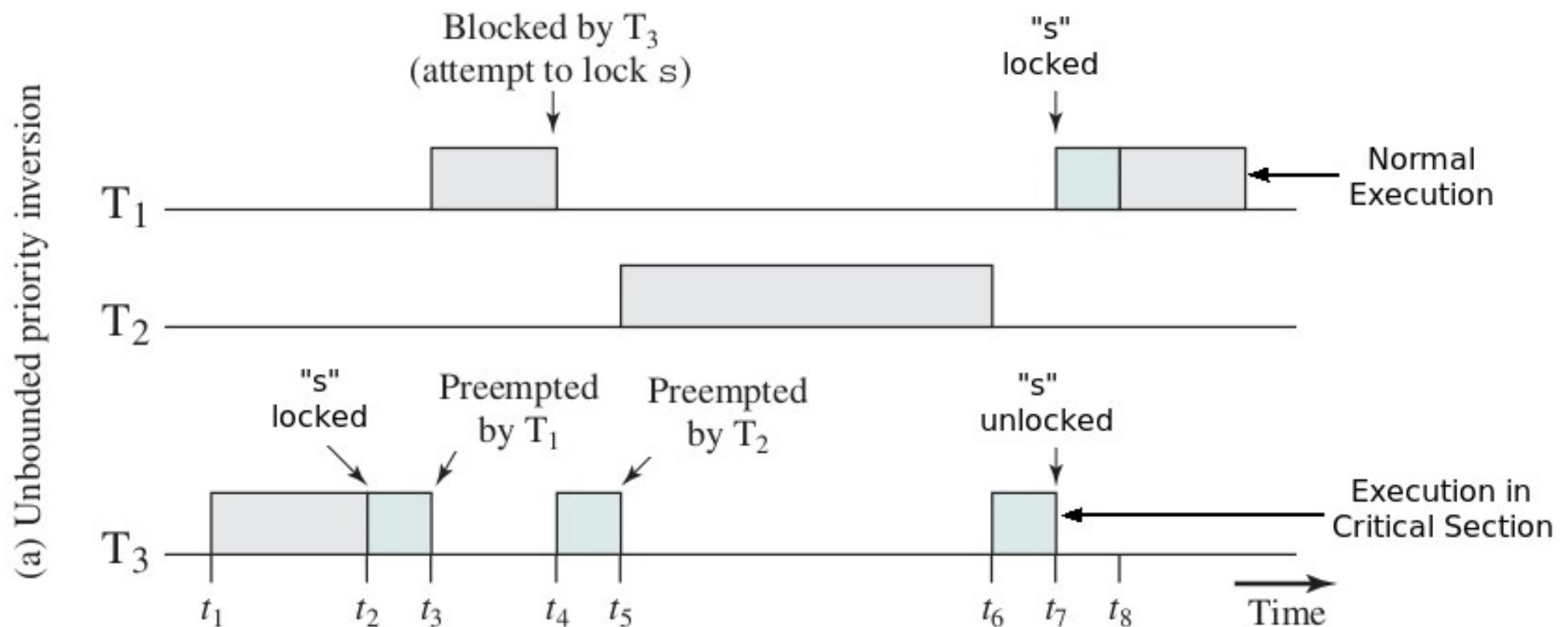
10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- Considerando que as tarefas T1 e T3 compartilham dados comuns e protegidos por um semáforo binário, seja a sequência eventos t_1 , t_2 , ... t_n que causou a inversão de prioridade.
-
- t_5 .. T#2 cuja prioridade é maior do que T#3, preempta T#3 e inicia a execução e, pode, em algum momento solicitar o mesmo recurso.
- t_6 .. T#2 está suspenso por razão não relacionada a T#1 e T#3, então, T#3 recupera a execução.
- t_7 .. T#3 deixa a sua região crítica e desbloqueia o semáforo, permitindo que T#1 preempta T#3, execute o semáforo para na sequência entrar na região crítica.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling ... 10.2.6 – Priority Inversion

- e.g., as tarefas T#1 e T#3 compartilham dados comuns e protegidos por um semáforo binário, seja a sequência eventos t_1, t_2, \dots, t_n que causou a inversão de prioridade.



10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- ... nestas circunstâncias, T#1 deve esperar por ambos T#3 e T#2 completarem seus ciclos de vida ou falharem na tentativa de reinicializar o temporizador antes que ele expire.
- 02 alternativas para evitar “**unbounded priority inversion**”:
- “**priority inheritance protocol**” .. tarefa de “menor” prioridade herda a prioridade da tarefa de “maior” prioridade que aguarda pelo recurso.
- .. mas sua prioridade volta ao valor que era tão logo o recurso seja liberado (neste caso, pela tarefa de menor prioridade).

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- ... nestas circunstâncias, T#1 deve esperar por ambos T#3 e T#2 completarem seus ciclos de vida ou falharem na tentativa de reinicializar o temporizador antes que ele expire.
- 02 alternativas para evitar “**unbounded priority inversion**”:
- “**priority ceiling protocol**” .. associa-se uma prioridade a cada recurso, que por sua vez é maior que a prioridade do usuário com a maior prioridade.
- ... escalonador dinamicamente atribui esta prioridade para qualquer tarefa que acesse o recurso e, uma vez que a tarefa utilize o recurso, sua prioridade retorna ao normal.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- “**priority inheritance**” resolve o problema da inversão de prioridade sem limitação como ilustrado na Fig. anterior:
- t1 – T#3 inicia execução;
- t2 – T#3 bloqueia o semáforo “s” e entra na região crítica;
- t3 – T#1 cuja prioridade é maior do que T#3, preempta T#3 e inicia a execução;
- t4 – T#1 tenta entrar na região crítica, mas é bloqueado pois o “s” já está bloqueado em T#3, assim T#3 recebe temporariamente a mesma prioridade de T#1 e recupera sua execução na sua região crítica;
- t5 – T#2 está pronto para executar, mas como T#3 tem prioridade maior, T#2 não está apto a preemptar T#3;
-

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

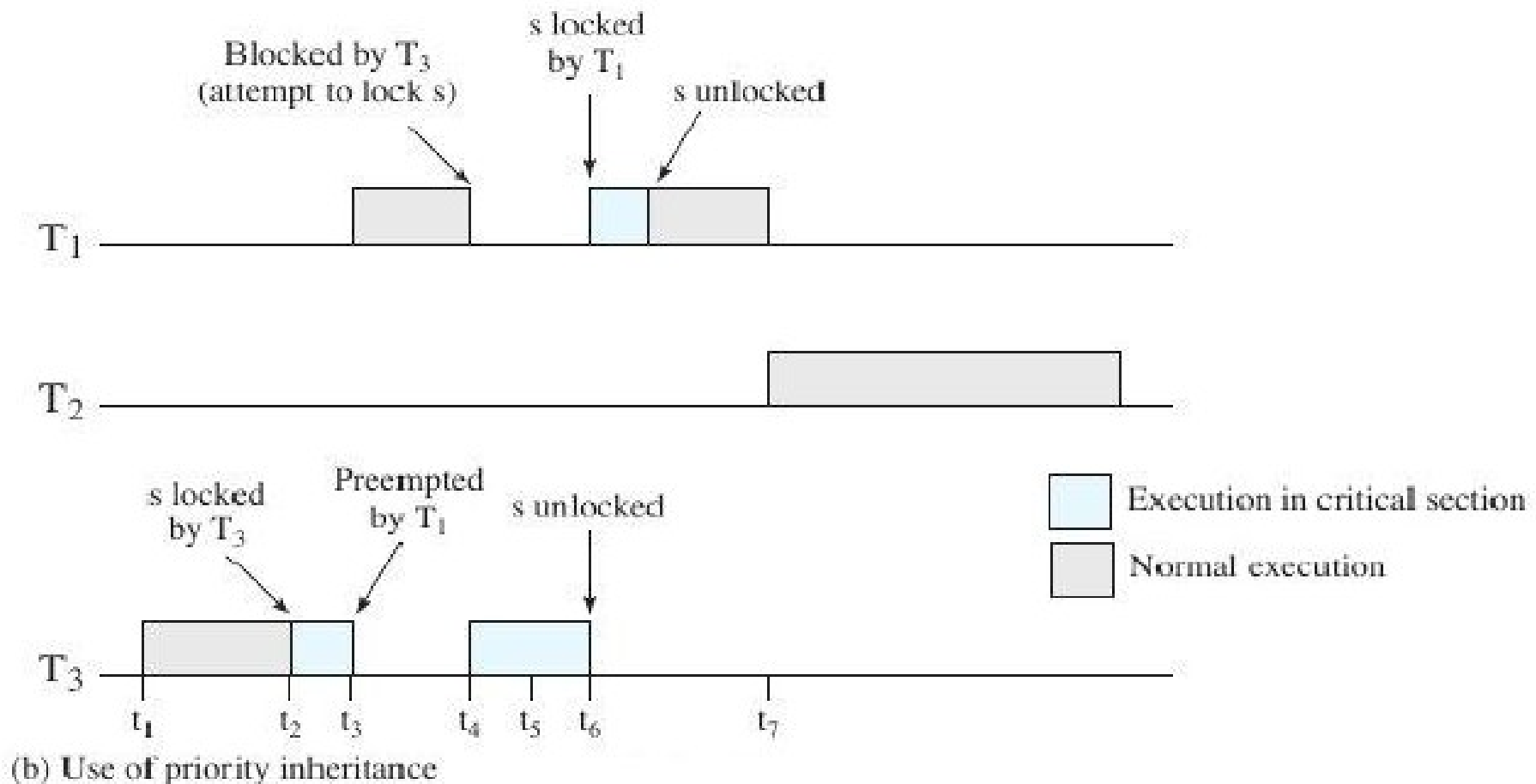
... 10.2.6 – Priority Inversion

- ... “priority inheritance” resolve o problema da inversão de prioridade sem limitação como ilustrado na Fig. anterior:
-
- t5 .. T#2 está pronto para executar, mas como T#3 tem prioridade maior, T#2 não está apto a preemptar T#3.
- t6 .. T#3 deixa a região crítica e desbloqueia o semáforo bem como tem sua prioridade alterada para a prioridade anterior, ou seja, T#1 preempta T#3, bloqueia o semáforo e entra na região crítica.
- t7 .. T#1 é suspenso por alguma razão não relacionada a T#2, e T#2 inicia execução.

10. SMP and RT Scheduling / 10.2. Real-Time Scheduling

... 10.2.6 – Priority Inversion

- “**priority inheritance**” resolve o problema da inversão de prioridade sem limitação como ilustrado na Fig. anterior.



10. SMP and RT Scheduling / 10.3 LINUX Scheduling

10.3 – LINUX Scheduling

- LEITURA COMPLEMENTAR (Operating Systems - Stallings – 7th)
- 10.3 – Escalonamento no Linux (Tópicos Avançados)
- 10.4 – Escalonamento no UNIX SVR4 (Tópicos Avançados)
- 10.5 - Escalonamento no Windows (Tópicos Avançados)

10. SMP and RT Scheduling / 10.4 LINUX Scheduling

10.4 – UNIX SRV4 Scheduling

- LEITURA COMPLEMENTAR (Operating Systems - Stallings – 7th)
- 10.3 – Escalonamento no Linux (Tópicos Avançados)
- 10.4 – Escalonamento no UNIX SVR4 (Tópicos Avançados)
- 10.5 - Escalonamento no Windows (Tópicos Avançados)

10. SMP and RT Scheduling / 10.5 WINDOWS Scheduling

10.5 – WINDOWS Scheduling

- LEITURA COMPLEMENTAR (Operating Systems - Stallings – 7th)
- 10.3 – Escalonamento no Linux (Tópicos Avançados)
- 10.4 – Escalonamento no UNIX SVR4 (Tópicos Avançados)
- 10.5 - Escalonamento no Windows (Tópicos Avançados)