

Introdução sobre Sistemas de Tempo Real, Escalonamento

Dr. Osmar Marchi dos Santos

Modelo de Tarefas

- É uma abstração que especifica as atividades que o sistema necessita para sua execução
 - Como o nome diz, é composto de tarefas (threads) que compõem as unidades de processamento
 - Cada tarefa tem um conjunto de informações associadas
- Diferentes modelos de tarefas existem na literatura, onde considera-se questões como, por exemplo:
 - Uso de arquiteturas diferentes (uni-processadores, multi-processadores)
 - Relações de precedência
 - Compartilhamento de informações

Modelo de Tarefas

- O modelo de tarefas permite utilizar técnicas de escalonamento para verificar se um sistema é escalonável (*todas as tarefas conseguem garantir suas restrições temporais*) ou não
- As tarefas são classificadas como:
 - Tarefas periódicas
 - Tarefas aperiódicas
 - Tarefas esporádicas

Modelo de Tarefas

- Tarefas periódicas: existe um período fixo (regular) para a execução da tarefa – e.g., tarefa que obtêm dados de um sensor de forma cíclica
- Tarefas aperiódicas: são tarefas não periódicas, que podem ocorrer de forma não-determinística (a qualquer momento) – e.g., tarefa que dispara um *air-bag* em um carro
- Tarefas esporádicas: são tarefas aperiódicas que possuem um “Tempo Mínimo de Chegada” (*Minimum Inter-arrival Time*), que impõem um limite na carga sobre o sistema – e.g., tarefa que obtêm dados de um sensor através de requisições

Modelo de Tarefas

- Outras características temporais da tarefa incluem:
 - Tempo de computação (*computation time* - C)
 - Pior tempo de execução (*worst-case execution time*) de uma tarefa
 - Calculado através de testes, simulação ou análise do programa (Worst-Case Execution Time Analysis)
 - Tempo de início (*start time* - st):
 - Indica o tempo inicial de ativação da tarefa
 - Tempo de término (*completion time* - ct)
 - Tempo em que a tarefa termina sua execução
 - Tempo de chegada (*arrival time* - a)
 - Instante em que o sistema toma conhecimento da ativação da uma tarefa
 - Tempo de liberação (*release time* - r)
 - Tempo em que a tarefa é incluída na fila de processamento do sistema para execução

Modelo de Tarefas

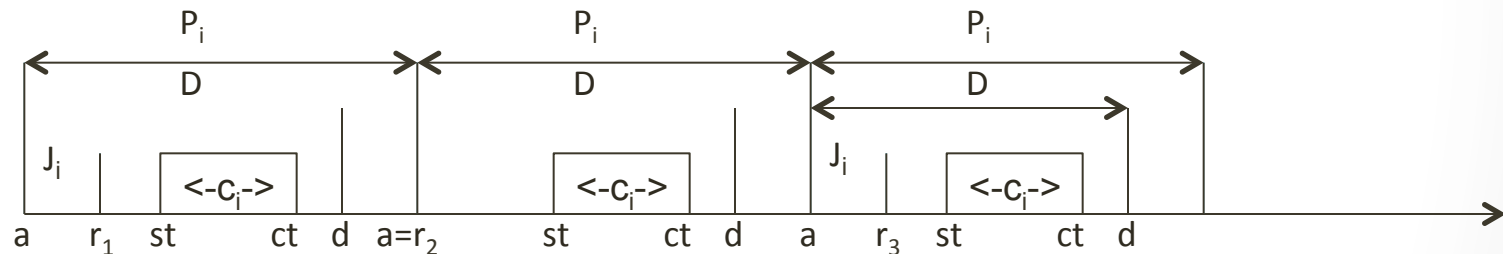
- *Release Jitter - J*
 - Uma vez que a tarefa chega no sistema, ela é brevemente liberada para inclusão na fila de processamento
 - Em alguns casos ela pode ser retardada (atrasada) por um determinado tempo, esse é o *Release Jitter*
 - Por exemplo, na espera (atraso) de uma mensagem de liberação para executar

Modelo de Tarefas

- Considerando essas características temporais, pode-se definir uma tarefa periódica T_i como uma quádrupla (C_i, P_i, D_i, J_i) onde:
 - C_i : Tempo de computação da tarefa i
 - P_i : Período de ativação da tarefa i
 - D_i : Deadline da tarefa i
 - J_i : Release Jitter da tarefa i
- D_i e J_i : são grandezas relativas (intervalos) medidas a partir do início do período P_i

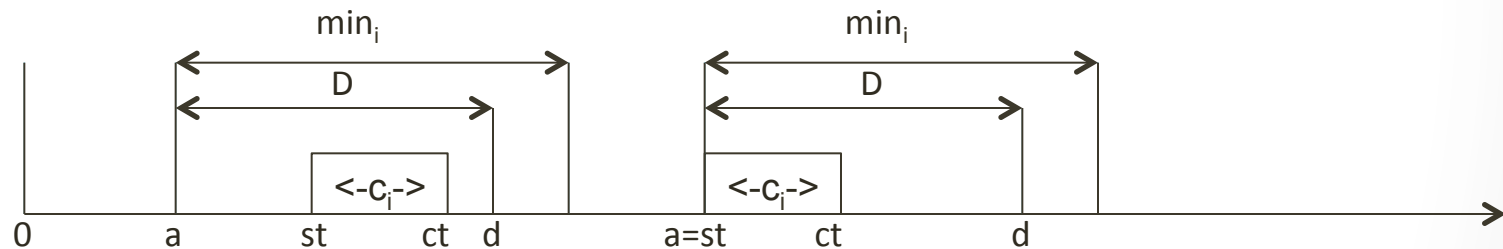
Modelo de Tarefas - Periódico

- O deadline absoluto (d) e o tempo de liberação (r) da $k^{\text{ésima}}$ ativação da tarefa periódica T_i são determinados a partir de períodos anteriores
 - $d_{ik} = (k-1)P_i + D_i$
 - $r_i = (k-1)P_i + J_i$ (pior caso possível)



Modelo de Tarefas - Esporádico

- Uma tarefa esporádica T_i é descrita como uma tripla (C_i, D_i, mit_i) onde:
 - C_i : Tempo de computação da tarefa i
 - D_i : Deadline da tarefa i
 - mit_i : Intervalo mínimo de tempo entre duas requisições



- Como calcular o deadline absoluto d ?

Discussão

- Pensem em dois exemplos para:
 - Tarefas Periódicas
 - Tarefas Aperiódicas
 - Tarefas Esporádicas
- Existe algum sistema de tempo real que poderíamos desenvolver em particular?

Escalonamento (*Scheduling*)

- Identifica o procedimento de ordenar tarefas que encontram-se disponíveis para execução no sistema
- O escalonador (*scheduler*) é o componente responsável por escolher qual a próxima tarefa a ser executada no processador
- A política de escalonamento define a forma que o escalonador irá escolher as próximas tarefas a serem executadas

Escalonamento (*Scheduling*)

- Principais abordagens de escalonamento:
 - Garantia em tempo de projeto (*off-line guarantee*):
 - Leva em consideração a carga do sistema e garante que no sistema existe reserva o suficiente para executar as tarefas considerando o pior caso
 - Garantia em tempo de execução (*on-line guarantee*):
 - Não é possível prever a carga do sistema, logo, testes são realizados durante a execução do sistema com o objetivo de garantir restrições temporais – tem de lidar com sobrecargas do sistema
 - Melhor esforço (*best-effort*):
 - Tentam encontrar uma escala do sistema sem testes ou com testes ainda mais fracos – mais usadas em sistemas de tempo real “soft”

Teste de Escalonamento

- Tem como objetivo verificar se a carga imposta pelo sistema irá conseguir ser executada pelo sistema, de forma que tarefas não percam seu(s) deadline(s)
 - Testes suficiente – bastante utilizado:
 - Conjuntos de escalonamento aceitos nesse teste são escalonáveis, porém um subconjunto descartado também pode ser escalonável
 - Testes necessários:
 - Descarta conjuntos de escalonamento não escalonáveis
 - Testes exatos (suficientes e necessários) – alta complexidade:
 - Identifica exatamente o conjunto de escalonamentos escalonáveis e não escalonáveis

Teste de Escalonamento

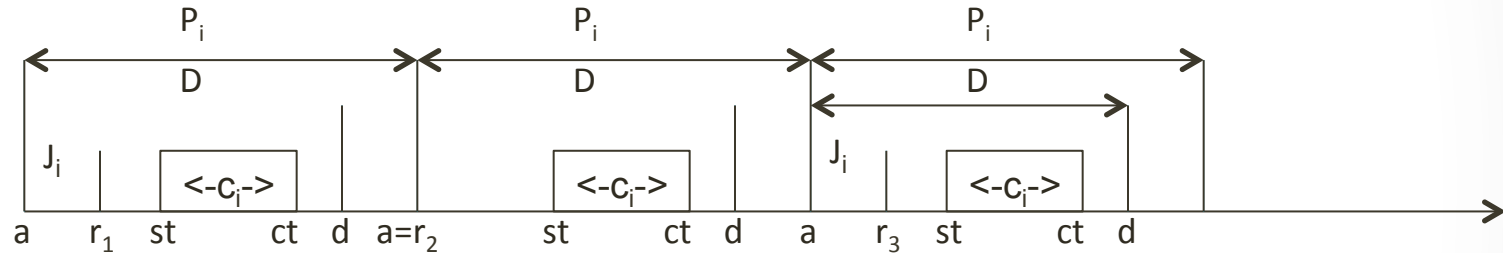
- Testes de escalonamento baseados em utilização são um exemplo de teste suficiente:
 - Utilização de uma tarefa: é a medida de ocupação de processador da tarefa, calculada como:
 - $U_i = C_i / P_i$
 - $U_i = C_i / \text{mit}_i$
- A utilização do processador é dada pelo somatório de todas as tarefas executadas:
 - $U = \sum_i^n U_i < 1$ (considerando um único processador)

Exercício

- Qual a utilização total (U) de uma aplicação composta por 2 tarefas periódicas onde:
 - Tarefa 1 – maior prioridade:
 - Computação = 5 ms
 - Período = 10 ms
 - Tarefa 2 – menor prioridade:
 - Computação = 6 ms
 - Período = 15 ms
- Esse sistema é escalonável?

Modelo de Tarefas - Periódico

- Possibilita determinar *a priori* a carga do sistema através dos tempos de ativação e computação (exemplo: $u_i = c_i / P_i$)



- Garantias em tempo de projeto sobre a “escalonabilidade” do conjunto de tarefas que compõem a aplicação

Escalonamento Baseado em Prioridades

- As prioridades das tarefas são derivadas “automaticamente” das restrições temporais das tarefas
 - Ao invés de utilizar outros atributos, como a importância de uma tarefa específica para o sistema
- Esse tipo de escalonamento apresenta melhor desempenho e flexibilidade
 - Por exemplo, comparado ao modelo executivo cíclico (*cyclic executive*)
- Discutiremos três abordagens de escalonamentos baseados em prioridades:
 - RM; EDF; DM