FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

T569 –SISTEMAS DE TEMPO REAL

Aula 8

Prof. Marcelo Sousa

Agenda

- Conceitos Importantes
- Tipos de *Tasks* de Tempo Real e Características
- Escalonamento de *Tasks*

CONCEITOS IMPORTANTES

Conceitos Importantes

- As *tasks* de tempo real produzem respostas a eventos gerados interna e/ou externamente.
  - Exemplo
    - Uma *task* é acionada sempre que o clock atinge uma determinada quantidade de *ticks* com o objetivo de ler a temperatura.
    - Uma *task* é acionada sempre que um botão é pressionado por um usuário

Conceitos Importantes

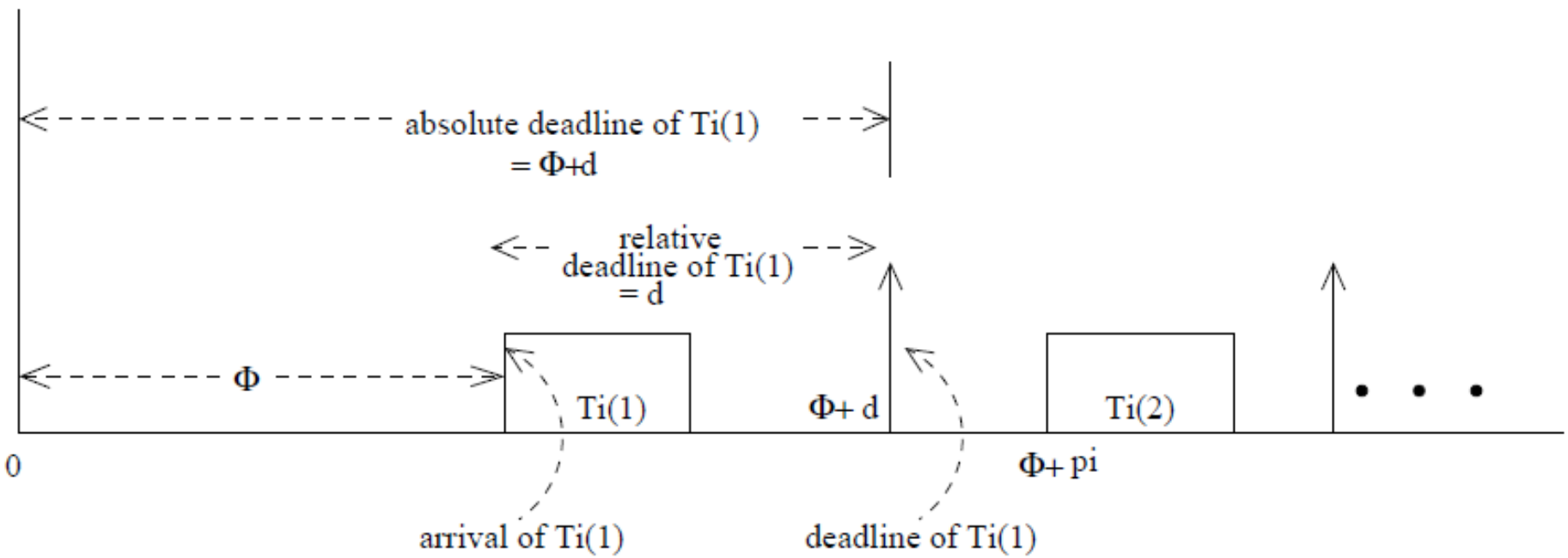
- *Task Instance*
  - Cada vez que uma *task* é criada, uma nova instância é gerada. Na 1ª chamada de uma *task*, é criada a sua 1ª instância. Na 2ª vez, uma 2ª instância é criada e assim, sucessivamente.
  - Definição:
    - Ti(j)
      - T → Task
      - i → Índice da taks
      - J → J-ésima instância da *task*

Conceitos Importantes

- Relative Deadline
  - É o intervalo entre o início e o prazo para o resultado daquela *task*.

- Absolute Deadline
  - É o valor de tempo absoluto a partir do instante 0s no qual o resultado daquela *task* é esperado.

Conceitos Importantes



Conceitos Importantes

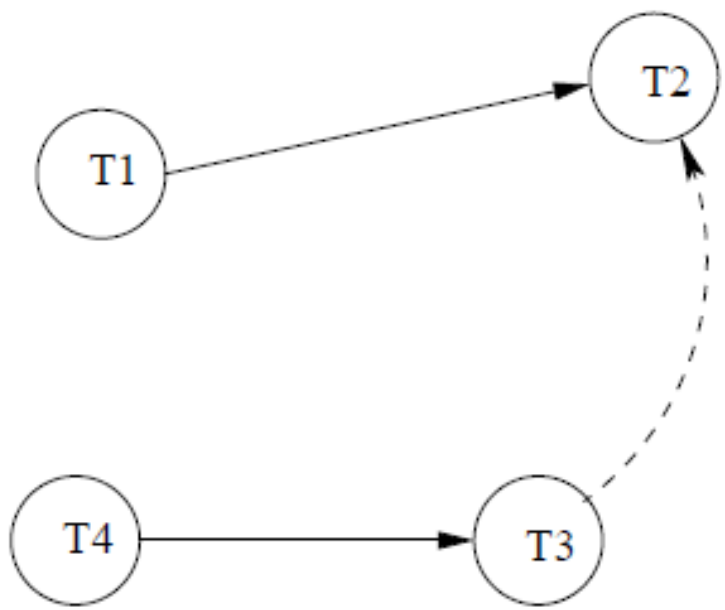
- Tempo de Resposta
  - É a duração do tempo entre o início do evento gerador daquela *task* e tempo para que ela gere resultados para o sistema.
- Precedência de Task
  - Uma *task* precedente é aquela que necessita obrigatoriamente ter sido concluída para que outra *task* seja iniciada.
  - Ti precede Tj, então $Ti_{(1)}$ precede $Tj_{(1)}$, $Ti_{(2)}$ precede $Tj_{(2)}$, ...

Conceitos Importantes

- Compartilhamento de dados
  - Precedência entre *tasks* geralmente implica em compartilhamento de dados entre as *tasks*.
  - Os dados geralmente necessitam ser compartilhados entre as *tasks* que possuem precedência

Conceitos Importantes

- Compartilhamento – Setas Tracejadas
- Precedência – Setas Preenchidas



TIPOS DE TASKS DE TEMPO REAL E SUAS CARACTERÍSTICAS

Tipos de Task

- *Task* Periódica
  - É uma *task* que se repete depois de um certo tempo.
  - A maioria das *tasks* em um sistema de tempo real são periódicas.
    - Verificação de sensores, cálculos de parâmetros, ...
  - Período: tempo fixo de intervalo no qual a tarefa se repete;
  - Fase: Tempo passado a partir do instante 0 até a execução da primeira instância da *task*

Tipos de Task

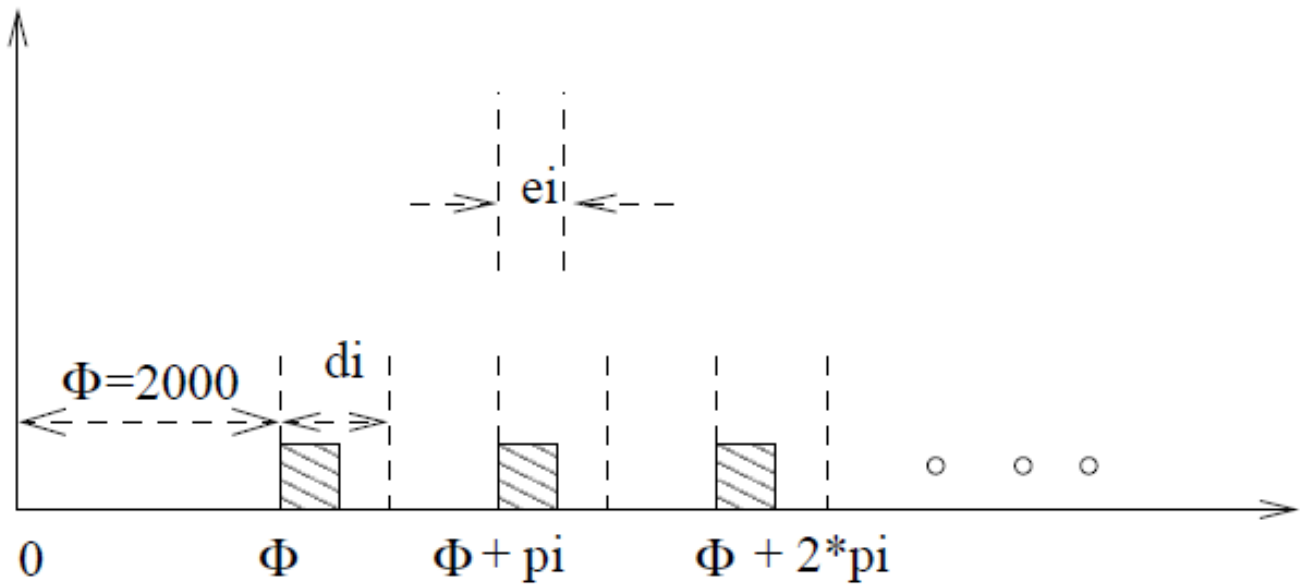
- Task Periódica

$$T_i = (\phi_i, p_{i,}, e_i, d_i)$$

  - ϕ_i Fase da *Task*
  - p_i Período da *Task*
  - e_i Pior tempo de execução daquela *Task*
  - d_i Deadline relativo da *Task*

Tipos de Task

- Task Periódica



Tipos de Task

- Task Esporádica
  - *Task* que se repete em instantes aleatórios.

$$T_i = (e_i, g_i, d_i)$$

  - e_i Pior tempo de execução daquela *task*
  - g_i Separação mínima entre duas *tasks* consecutivas
  - d_i Deadline relativo da *Task*

Tipos de Task

- Task Esporádica
  - A separação mínima (g_i) entre duas instâncias consecutivas da *task* implica que em caso de acontecimento de uma instância, outra instância não poderá ser criada antes do tempo g_i.
  - Algumas *tasks* esporádicas possuem criticidade bastante elevada
    - Ex.: Chegada de mensagens de emergência.
  - O momento em que estas *tasks* irão ocorrer não pode ser previsto.

Tipos de Task

- Task Aperiódica
  - *Task* bastante semelhante à *task* esporádica.
  - No entanto, o tempo *gi* pode ser zero, ou seja, as tasks aperiódicas podem ser instanciadas no mesmo instante de tempo.
  - O *deadline* de uma task aperiódica é dado por um valor médio ou estatístico.
  - *Tasks* aperiódicas são geralmente do tipo *soft-real time*.

Tipos de Task

- Task Aperiódica
  - *Tasks* aperiódicas podem se repetir de maneira rápida sucessivamente. Então, é possível que alguns deadlines não sejam atendidos. Como discutido anteriormente, uma *soft real-time task* pode aceitar a perda de alguns *deadlines*
  - Exemplo:
    - Ato de pressionar as teclas do keyboard, movimentos do mouse, solicitações do operador. Ou seja, todos os comandos de interação com o sistema são *tasks* aperiódicas

ESCALONAMENTO DE TAREFAS

Escalonamento de Tarefas

- Definição:
  - É uma ferramenta utilizada para determinar a ordem na qual a *task* será executada pelo sistema operacional
  - Todo sistema operacional depende de um ou mais escalonadores de tarefas para preparar a ordem de execução das *tasks*
  - Cada escalonador de tarefas é caracterizado pelo algoritmo de escalonamento utilizado.

Escalonamento de Tarefas

- Escalonamento Válido (*Valid Schedule*)
  - É um escalonamento onde no máximo uma *task* é assinalada para o processador por vez
  - Nenhuma tarefa é escalonada antes do seu tempo de execução
  - Precedências e restrições de recursos de todas as *tasks* são satisfeitas.

Escalonamento de Tarefas

- Escalonamento Praticável (*Feasible Schedule*)

  – Um **escalonamento válido** é chamado de **escalonamento praticável**, apenas se todas as *tasks* atingirem suas **limitações de tempo** no escalonador.

Escalonamento de Tarefas

- ***Proficient Scheduler***
  - O escalonador de tarefas *sch1* é dito mais proficiente que outro escalonador *sch2*:
    - Se *sch1* viabilizar o escalonamento de um conjunto de tarefas que o escalonador *sch2* ***não*** conseguiu viabilizar, mas não vice-versa.
    - Ou seja, que exista pelo menos uma tarefa no conjunto de tarefas escalonáveis de *sch1* que o *sch2* não possa escalonar.
  - Caso os dois escalonadores possam viabilizar o escalonamento de um conjunto de *tasks*, eles são ditos **escalonadores igualmente proficientes**

Escalonamento de Tarefas

- ***Optimal Scheduler***
  - Um escalonador de tarefas de tempo real é dito ótimo se ele viabilizar o escalonamento de tarefas que qualquer outro escalonador possa escalonar.
  - Em outras palavras, não é possível encontrar outro escalonador que seja mais proficiente que um escalonador ótimo;
  - Se um **escalonador ótimo** não conseguir viabilizar o escalonamento de uma tarefa, nenhum outro escalonador poderá ser capaz de escaloná-la.

Escalonamento de Tarefas

- ***Scheduling Points***
  - São pontos na linha do tempo no qual o escalonador toma as decisões sobre qual tarefa será executada em seguida.
  - O escalonador não será executado todo o tempo pelo sistema operacional. Ele é ativado apenas nos **pontos de escalonamento** para tomar a decisão a respeito de qual tarefa será executada em sequência.
  - **Clock Driven** → definidos por instantes de tempo.
  - **Event Driven** → definidos pela ocorrência de certos eventos.

Escalonamento de Tarefas

- ***Preemptive Scheduler***
  - Quando uma *task* de maior prioridade chega para ser atendida, suspende a execução da tarefa de menor prioridade, executando a *task* de maior prioridade.
  - Em um escalonador **preemptivo**, não é possível que uma *task* de maior prioridade esteja em espera para execução e uma tarefa de menor prioridade esteja sendo executada.

Escalonamento de Tarefas

- Utilização
  - A utilização do processador por uma *task* é o tempo médio no qual aquela tarefa é executada por unidade de tempo.

$$u_i = \frac{e_i}{p_i}$$

u_i – Utilização

e_i – Tempo de execução

p_i – Período da task T_i

Escalonamento de Tarefas

- Utilização
  - A utilização total (U) de um conjunto de *tasks* {*Ti*} é dada por:

$$U = \sum_{i=1}^{n} \frac{e_i}{p_i}$$

  - O objetivo de um bom escalonador é viabilizar o escalonamento mesmo que o conjunto de tarefas possua utilização bastante elevada, ou seja, utilização próxima de 1.
  - Em um sistema uniprocessador, não é possível escalonar conjuntos de tarefas que possuam utilização > 1.

Escalonamento de Tarefas

- Jitter
  - É o desvio no tempo da *task* de seu comportamento periódico.
  - Jitter de inicio da *task*: é o desvio no tempo de início esperado para execução daquela *task*
    - Pode ser causado por clocks imprecisos e congestionamentos na rede.
  - Jitter de libração da *task*: É o desvio de tempo para liberação daquela *task*
    - Pode ser causado por algoritmos de escalonamento que priorizam a finalização de outras *tasks* em detrimento de algumas.

Escalonamento de Tarefas

- Classificação de Algoritmos

  1) Os algoritmos podem ser classificados de acordo com a definição do ponto de escalonamento.

  - Clock Driven
    - Pontos de escalonamento determinados por interrupções recebidas do clock
  - Event Driven
    - Pontos de escalonamento determinados por eventos que causam interrupções no sistema
  - Hybrid
    - Pontos de escalonamento determinados das duas maneiras anteriores.

Escalonamento de Tarefas

- Classificação de Algoritmos
  - Clock Driven
    - São algoritmos mais simples, mas são eficientes.
    - São mais empregados em ambientes embarcados que em ambientes tradicionais
    - Exemplo.:
      - Table-driven
      - Cyclic

Escalonamento de Tarefas

- Classificação de Algoritmos
  - Event Driven
    - São escalonadores mais elaborados.
    - São escalonadores geralmente mais proficientes e flexíveis que os baseados em clock
    - Podem escalonar tarefas esporádicas e aperiódicas
    - Exemplo:
      - Simple Priority-based
      - Rate Monotonic Analysis (RMA)
      - Earliest Deadline First (EDF)

Escalonamento de Tarefas

- Classificação de Algoritmos

  2) Outra forma de classificar os algoritmos de escalonamento é baseada no tipo de testes de aceitação.

  - *Planning Based*
    - Utiliza planos de execução.
  - *Best Effort*
    - Nenhum teste de aceitação é realizado. Sempre que uma task chega para escalonamento, ela é escalonada para execução. Nenhuma garantia é dada para o atendimento do *deadline*.

Escalonamento de Tarefas

- Classificação de Algoritmos

  3) Uma terceira opção de classificação é baseada no alvo onde serão processadas as tarefas

  - Uniprocessor
    - Classificação mais simples dos algoritmos
  - Multiprocessor
    - O algoritmo deve gerenciar qual processador executará qual tarefa
  - Distribuído
    - Várias unidades de processamento realizam o escalonamento de *tasks*. A comunicação entre as unidades de processamento devem trocar poucas informações através da rede para evitar congestionamento

Próxima Aula

- Prática: Remoção de Tarefas e Escalonadores