

Sistemas de Tempo Real: Executivo Cíclico

Rômulo Silva de Oliveira
Departamento de Automação e Sistemas - DAS – UFSC

romulo.deoliveira@ufsc.br
<http://www.romulosilvadeoliveira.eng.br>

Setembro/2015

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 1

Executivo Cíclico – Introdução

- Também chamado de
 - Time-driven scheduler ou Clock-driven scheduler
- Indicado para tarefas com deadline hard
- Indicado para tarefas periódicas
- Parâmetros das tarefas precisam ser conhecidos a priori
- Parâmetros das tarefas não mudam durante a execução
- Tabela descrevendo a escala de execução é criada no projeto
 - Durante a execução esta escala é repetida ciclicamente
- Garantia da escalonabilidade está na inspeção da tabela
 - Se todos os deadlines são cumpridos, sistema é escalonável

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 2

Executivo Cíclico – Introdução

- Todo o trabalho de escalonamento é feito em projeto
- Resultado é uma **grade de execução** (*time grid*)
 - Grade determina qual tarefa executa quando
 - Garantia obtida através de uma simples inspeção da escala
- Durante a execução:
 - Pequeno programa lê a grade e dispara a tarefa apropriada (**executivo**)
 - Quando a grade termina ela é novamente repetida (**cíclico**)
- É a forma tradicional de implementar sistemas de tempo real críticos
- Muito usado em aplicações embutidas (Embedded Systems)
- **Vantagem:** Comportamento completamente conhecido
 - Determinismo da escala de execução
- **Desvantagem:** Escalonamento muito rígido, tamanho da grade

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 3

Executivo Cíclico – Exemplo

Tarefa Ti	Período Pi	Tempo de computação Ci
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

Mínimo múltiplo comum dos períodos = 100

Ciclo maior = 100 ms

Máximo divisor comum dos períodos = 5

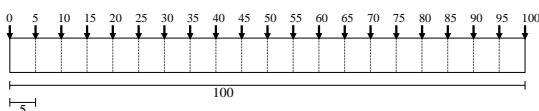
Ciclo menor = 5 ms

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 4

Executivo Cíclico – Exemplo

Tarefa Ti	Período Pi	Tempo de computação Ci
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

Interrupção do timer

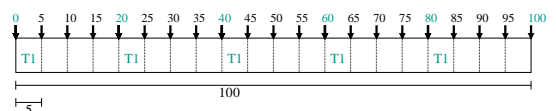


Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 5

Executivo Cíclico – Exemplo

Tarefa Ti	Período Pi	Tempo de computação Ci
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

Interrupção do timer

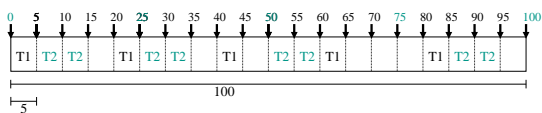


Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 6

Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

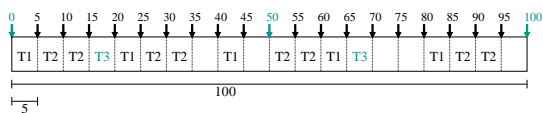
Interrupção do timer



Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

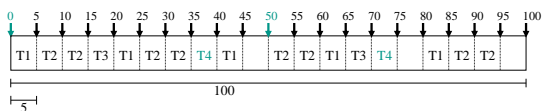
Interrupção do timer



Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

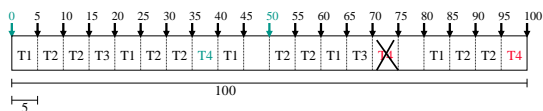
Interrupção do timer



Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

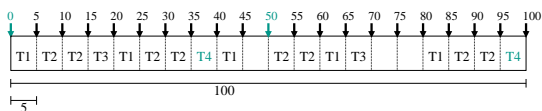
Interrupção do timer



Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

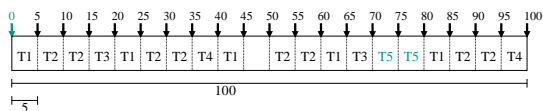
Interrupção do timer



Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	20 ms	5 ms
T2	25 ms	8 ms
T3	50 ms	5 ms
T4	50 ms	4 ms
T5	100 ms	10 ms

Interrupção do timer



Executivo Cíclico – Construção da Escala

- Escalonabilidade verificada pela simples inspeção da escala
- Tamanho da grade
 - MMC dos períodos cresce rapidamente para números quaisquer
 - Engenheiros tentem a usar períodos múltiplos entre si
 - Possível “ajustar” períodos, diminuindo o valor, em algumas aplicações
- Com solução preemptiva é mais fácil de escalonar
- Mas solução não preemptiva elimina o problema de seção crítica
- Solução de compromisso:
 - Permitir preempção de T1 apenas por tarefas com as quais T1 não compartilha seção crítica
 - Facilita a escalonabilidade (mais opções de escalas válidas)
 - Dificulta a construção da escala (existe uma restrição extra)

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 13

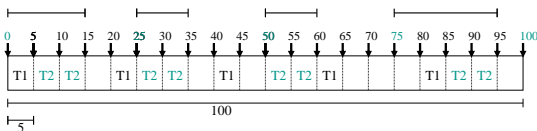
Executivo Cíclico – Construção da Escala

- Tamanho do ciclo menor:
 - Quanto menor, maior será o overhead (tratador de interrupção do timer)
 - Quanto maior, mais sobras existirão para algumas tarefas
 - No exemplo, tarefa T2 precisa de 8ms mas foram alocados 10ms para ela
 - Permite detectar falta (tarefa excede seu worst-case execution time)
- Pode ser feito para multiprocessador
 - Apenas o problema de construir a escala fica mais complexo
- Restrições devem ser observadas na construção da grade
 - Períodos, tempos máximo de computação
 - Precedências, exclusões mútuas
 - Jitter de saída

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 14

Executivo Cíclico – Construção da Escala

- Tempo entre chegada e conclusão varia para T2: Jitter de saída
- Algumas aplicações não toleram jitter de saída grande
- Neste caso:
 - Existe um objetivo secundário na construção da escala
 - Minimizar o jitter de saída das tarefas



Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 15

Executivo Cíclico – Construção da Escala

- Executivo cíclico é em essência **time-triggered**
- Como comportar tarefas esporádicas ?
 - Podem chegar a qualquer momento
- Solução é reservar espaço de tempo em tempo para a possibilidade da tarefa esporádica chegar
 - Viável quando intervalo mínimo entre chegadas for igual ao deadline
 - Mas subutiliza o processador (caso ela não chegue)
 - Pode ser impossível se o deadline for muito apertado, o que é comum para tarefas de emergência ou exceção
- Exemplo de esporádica no limite:
 - I=100ms, C=10ms e D=10ms
 - 100% do processador será reservado para esporádica com utilização 10%

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 16

Executivo Cíclico – Construção da Escala

- Determinação da escala é um problema de otimização
- Solução ótima apenas para sistemas pequenos
 - poucas dezenas de tarefas
- Existem várias heurísticas sub-ótimas na literatura
- Meta-heurísticas podem ser usadas
 - Algoritmos genéticos
 - Simulated annealing
 - Busca tabu
 - Etc

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 17

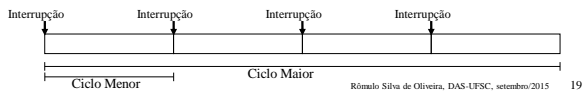
Executivo Cíclico – Implementação Simplificada

- O design do software é concorrente
- A implementação pode ser através de
 - Um conjunto de procedimentos
- Procedimentos são mapeados sobre um conjunto de ciclos menores (**minor cycles**) que formam juntos a escala completa ou ciclo maior (**major cycle**)
- O Ciclo Menor determina o mínimo ciclo de tempo possível
- O Ciclo Maior determina o máximo ciclo de tempo possível
- É alocado tempo para o WCET de cada tarefa a cada período dela
- Não existe preempção (exclusão mútua assegurada)

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 18

Executivo Cíclico – Implementação Simplificada

- Toda a sequência de execução é repedita a cada Major Cycle (Ciclo Maior)
 - Todas as execuções de Ciclo Maior são iguais
- O Ciclo Maior é dividido em uma sequência de Minor Cycles (Ciclo Menor)
 - Ciclos Menores podem ser diferentes entre si
- Interrupção do Timer indica início de cada Ciclo Menor
- Decisões de escalonamento somente no início de cada Ciclo Menor



Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 19

Executivo Cíclico – Implementação Simplificada

- Duração do Ciclo Maior indica instante no tempo equivalente ao instante inicial da tabela
- É possível voltar ao início da tabela e repetir tudo novamente
- Valor natural para a duração do Ciclo Maior (L): Mínimo Múltiplo Comum dos períodos das tarefas

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 20

Executivo Cíclico – Implementação Simplificada

- Ciclo menor (f) precisa ser longo o suficiente para permitir que todos os jobs terminem sua execução dentro dele
 - Evita preempções
 - Evita chaveamento de contexto
 - Fornece exclusão mútua
$$f \geq \max_{1 \leq i \leq n} (C_i)$$
- Conveniente que todos os períodos de tarefa sejam números múltiplos da duração do Ciclo Menor (f)
 - Interrupção do timer a cada Ciclo Menor
$$\left\lfloor \frac{P_i}{f} \right\rfloor - \frac{P_i}{f} = 0$$
- Várias tarefas podem ser executadas dentro de um mesmo Ciclo Menor

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 21

Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	25	10
T2	25	8
T3	50	5
T4	50	4
T5	100	2

Mínimo múltiplo comum dos períodos = 100
Máximo divisor comum dos períodos = 25

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 22

Executivo Cíclico – Exemplo

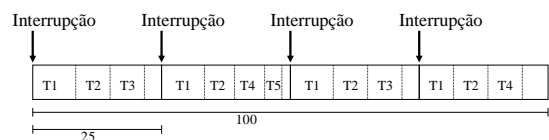
```

while( true ) {
    wait_for_interrupt;
    funcao_t1(); funcao_t2(); funcao_t3();
    wait_for_interrupt;
    funcao_t1(); funcao_t2(); funcao_t4(); funcao_t5();
    wait_for_interrupt;
    funcao_t1(); funcao_t2(); funcao_t3();
    wait_for_interrupt;
    funcao_t1(); funcao_t2(); funcao_t4();
}
    
```

Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 23

Executivo Cíclico – Exemplo

Tarefa T_i	Período P_i	Tempo de computação C_i
T1	25	10
T2	25	8
T3	50	5
T4	50	4
T5	100	2



Rômulo Silva de Oliveira, DAS-UFSC, setembro/2015 24

Executivo Cíclico – Implementação Simplificada

- Tabela indica que tarefas fazem parte de cada Ciclo Menor
 - Para toda a duração do Ciclo Maior
- Existem conjuntos de tarefas para os quais é impossível aplicar a solução clássica
 - Necessário adaptar as tarefas
- Tarefas aperiódicas podem ser executadas dentro de cada Ciclo Menor depois que todas os jobs garantidos alocados àquele Ciclo Menor já foram concluídos
 - Executam até a próxima interrupção do timer (novo Ciclo Menor)

Executivo Cíclico – Implementação Simplificada

- Overhead de escalonamento no início de cada Ciclo Menor
- Se ocorreu overrun do Ciclo Menor anterior, ação é necessária
- Job interrompido era aperiódica não garantida
 - Simplesmente preempta o job
- Job interrompido era periódico garantido mas não crítico
 - Preempta o job, sinaliza ocorrência de uma falta temporal
- Job interrompido era periódico garantido e crítico
 - Grave falta temporal
 - Continua executando este job para garantir consistência dos dados
 - Espera que folga do próximo Ciclo Menor resolva o problema
 - Dispara tratamento de exceção, pânico

Executivo Cíclico – Implementação Simplificada

- Pode ser desenvolvido a mão sem nenhum método especial
 - No caso de sistemas pequenos/simples
- Dado que a tabela de execução é construída em projeto
 - Algoritmos complexos também podem ser usados
 - Trata-se de um problema de otimização com complexidade exponencial
 - Soluções ótimas são possíveis até um certo tamanho de sistema
 - Para sistemas grandes são usadas heurísticas (sub-ótimas)
 - Uso de meta-heurísticas é popular (busca tabu, genético, simulated annealing)
- Objetivos secundários podem existir
 - Distribuir as folgas uniformemente para favorecer aperiódicas
 - Reduzir jitter de saída das tarefas

Executivo Cíclico – Implementação Simplificada

- Nenhum processo precisa realmente existir em tempo de execução
 - Cada Ciclo Menor pode ser apenas uma sequência de chamadas de subrotinas
 - Não é necessário implementar multiprogramação
- Subrotinas podem compartilhar um espaço de endereçamento comum
 - Podem passar dados entre eles
 - Esses dados não precisam ser protegidos (via um semáforo, por exemplo) porque o acesso concorrente não é possível
- Todos os períodos das “tarefas” devem ser múltiplos do tempo de ciclo menor

Executivo Cíclico – Implementação Simplificada

- Dificuldade em incorporar processos com períodos longos
- Qualquer “tarefa” com tempo de computação maior precisará ser dividido em um número fixo de procedimentos com tamanho mediano
 - Isto pode prejudicar a estrutura do código, sendo mais sujeito a bugs
- Métodos mais flexíveis de escalonamento são difíceis de suportar

Executivo Cíclico – Resumo

- Executivo cíclico é menos flexível porém oferece determinismo de escala
- Vantagens
 - É a forma tradicional para sistemas críticos
 - Comportamento completamente conhecido
 - Fácil detectar qualquer falha de projeto
 - Adequado para tarefas periódicas, as quais são maioria nos sistemas críticos
- Desvantagens
 - Não lida bem com tarefas que não são periódicas
 - Tabela pode ficar grande, caso períodos não sejam múltiplos entre si
 - No caso de WCET mal calculado, o que fazer ?
 - Tarefas muito longas precisam ser quebradas em várias sub-tarefas
 - O executivo cíclico é difícil de construir e manter — é um problema NP-hard
- Na realidade,
 - Determinismo da escala de execução não é necessário
 - Necessário é previsibilidade quanto ao cumprimento dos deadlines