

Clock Driven Scheduling

Sérgio Campos

Clock Driven Scheduling

- Restrições
- Assíncrono X síncrono
- Cyclic executive
- Slack stealing
- Jobs esporádicos

Restrições

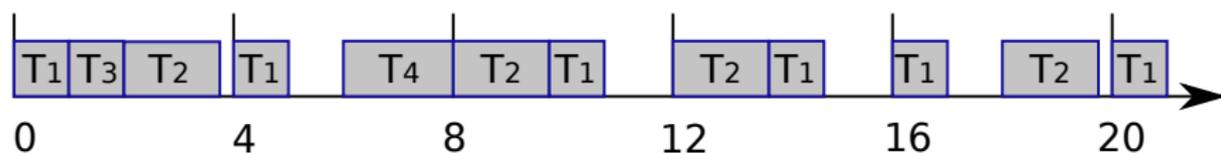
O sistema tem que ser:

- Determinístico;
- Número fixo de processos;
- Parâmetros conhecidos;
- Periódicos ou aperiódicos;

Escalonamento estático:

- Off-line;
- Garantido;
- Pode usar algoritmos de escalonamento sofisticados e caros.

Escalonamento Assíncrono

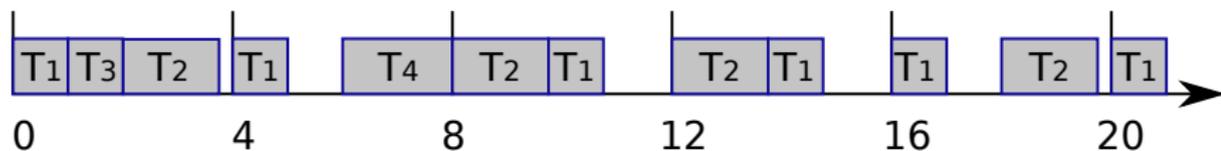


- $T_1 = (4, 1)$, $T_2 = (5, 1.8)$, $T_3 = (20, 1)$, $T_4 = (20, 2)$
- $U_1 = .25$; $U_2 = .36$; $U_3 = .05$; $U_4 = .1$
- Hiperperíodo = 20

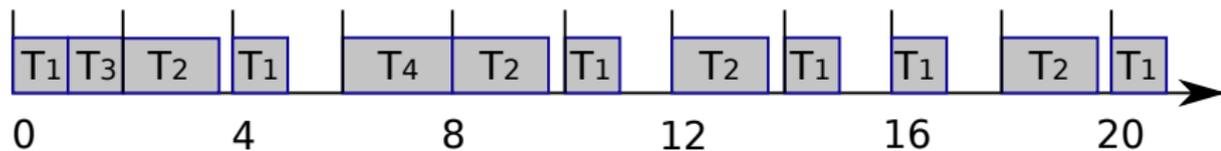
Escalonamento Síncrono

Tempos de decisão são chamados frames.

- Exemplo anterior:
Assíncrono



Síncrono:



- $T_1 = (4, 1)$, $T_2 = (5, 1.8)$, $T_3 = (20, 1)$, $T_4 = (20, 2)$

Cyclic Executive

Variação do método anterior aonde:

- Toda decisão de escalonamento ocorre no início da frame.

No início da frame o CE:

- Checa se houve overrun;
- Decide deterministicamente quem vai executar nesta frame;
- Cada job recebe um slice.

Após todos os jobs periódicos, jobs aperiódicos executam.

Tipo de round-robin temporizado.

Cyclic Executive

- Dividimos a linha do tempo em *quadros* ou *frames*.
- Cada tarefa é executada como uma chamada de procedimento que cabe dentro do quadro.
- Múltiplas tarefas podem ser executadas em um quadro, mas a tabela só é checada a cada início de quadro.
- Além de decidir quem executa o escalonador pode monitorar o sistema e checar:
 - Se algum processo perdeu o deadline (overrun);
 - ...

Cyclic Executive

E se não for possível conciliar o tamanho da frame com os tempos de execução ?

- Por exemplo, os deadlines forçam a frame ser < 10
- Mas tem uma tarefa que gasta tempo 15 para executar.
- Solução: quebrar o tarefa em três subtarefas.

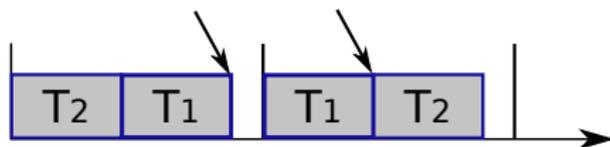
Tempo de Resposta de Aperiódicos

Melhor executar:

- O mais cedo possível ?
- O mais tarde possível, mas dentro do prazo ?

Não há vantagens em se executar antes do necessário... Preguiça é uma virtude???

Portanto:



Tempo de Resposta de Aperiódicos

Ruim: rodam depois dos periódicos, e se der tempo.

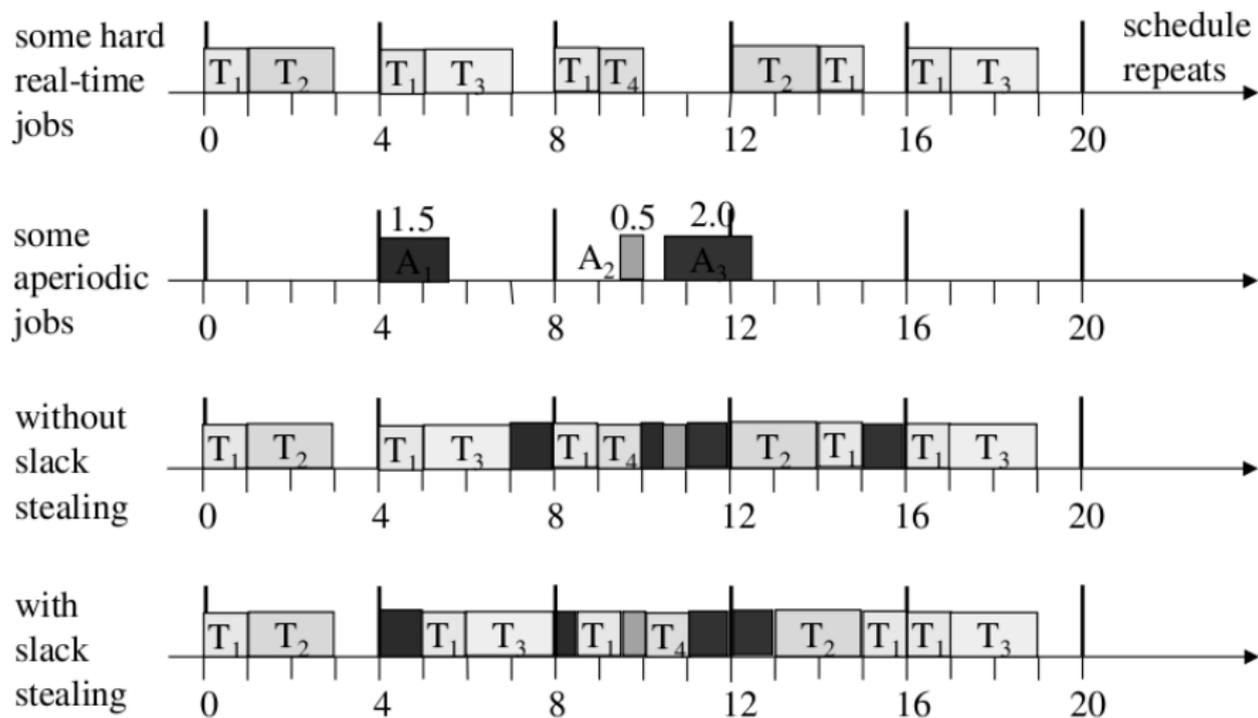
Bem, se der tempo não tem como mudar, mas podem rodar antes dos periódicos:

- Não há vantagem em terminar jobs periódicos antes do tempo.
- Slack stealing — slack = folga

Problema:

- Execução tem que ser interrompida quando o slack acabar;
- Granularidade: se a frame for 10ms, não existe temporização menor para avisar que o slack era de 2.3ms.

Slack Stealing Example



Jobs Esporádicos

São jobs aperiódicos que têm hard deadlines.

- Pisar no freio!
- *incoming missile...*

Problema: não há garantias a priori de execução.

Teste de Aceitação ou **Acceptance test**: A cada release o escalonador decide se o novo job pode ser aceito

- Job pode ser aceito se houver slack.
- Se não houver, é melhor recusar agora enquanto existe tempo para recuperação.
 - Exemplo: robô pode ser chamado para pegar um objeto próximo;
 - Contudo, se o job para pegá-lo for recusado, o robô tem que parar!
 - Job para parar é mais simples e é aceito.
 - Mudança de modo: apaga job antigo, submete job novo, reescalona.

Jobs Esporádicos

Como escalonar esporádicos ?

- Escalonados como aperiódicos levando em conta seu deadline
- Se houver vários esporádicos: usar EDF para escolher entre eles;
- Lembrar que EDF deixa de ser ótimo se escalonamento ocorrer somente no início da frame.

Outras técnicas existem que serão vistas depois: *servidores de esporádicos*.

Vantagens e Desvantagens do Cyclic Executive

Vantagens:

- Tecnologia KISS: Keep it Simple and Straightforward...
- Não há necessidade de controle de processos, concorrência, ...
- Muito confiáveis
- Muito usados em aplicações críticas

Desvantagens:

- Qualquer modificação pequena exige refazer tudo!
- Release times são fixos
- Sincronização.