

Escalonamento

Escalonamento em Sistemas Operacionais Embarcados de Tempo Real

Escalonamento

- Tipos de Escalonamento
 - Preemptivo
 - Não-Preemptivo

Escalonamento em RTOS

- Duas preocupações específicas:
 - Manter os prazos de tempo de resposta
 - Previsibilidade - Garantir que respostas sejam entregues de maneira ordenada e correta em tempo

Escalonamento em RTOS

- Rate Monotonic Scheduling (RMS)
 - Preemptivo
 - Prioridades fixas
 - Prioridades dependentes de tempo da frequência que o processo se apresenta
 - Aplicável apenas a processos periódicos

Escalonamento em RTOS

- Earliest Deadline First (EDF)
 - Preemptivo
 - Considera o momento que a resposta deve ser entregue
 - Quanto mais próximo do deadline mais prioritário o processo
 - Aplicável a processos periódicos ou aperiódicos
 - Algoritmo mais complexo

Alguns RTOS para embarcados

- QNX
- Linux
 - uCLinux
 - eCos
 - SnapGear
- NetBSD
- MicroC/OSII

QNX

- Direcionado a plataformas ARM e X86
- O algoritmo de escalonamento é definido por thread, não globalmente
- Múltiplos Algoritmos de Escalonamento
 - FIFO
 - Round Robin
 - Sporadic
 - Adaptative Partitioning Scheduler

FIFO

- First in First Out
 - Comum em sistemas batch
 - Executa o processo “até o fim” antes de passar o processador a outro processo
- Um processo fica de posse da CPU até que:
 - Termine sua execução e libere espontaneamente
 - Fique Bloqueado
 - Um processo de maior prioridade esteja pronto

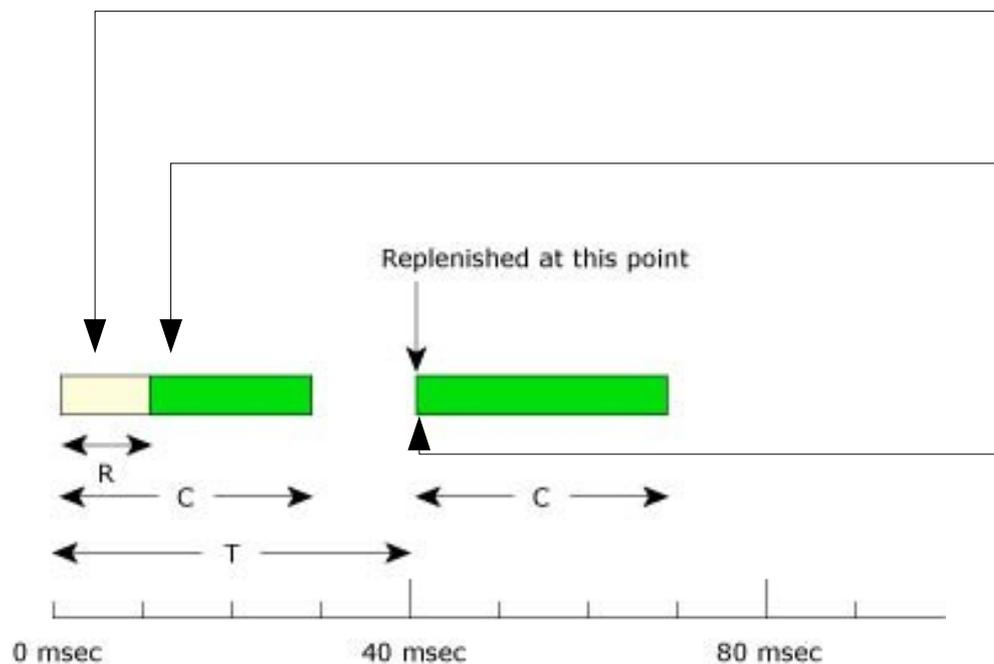
Round Robin

- Um processo fica de posse do processador até que:
 - Termine sua execução e libere espontaneamente
 - Fique Bloqueado
 - Um processo de maior prioridade esteja pronto
 - **Expire o seu tempo de execução (quantum)**

Sporadic

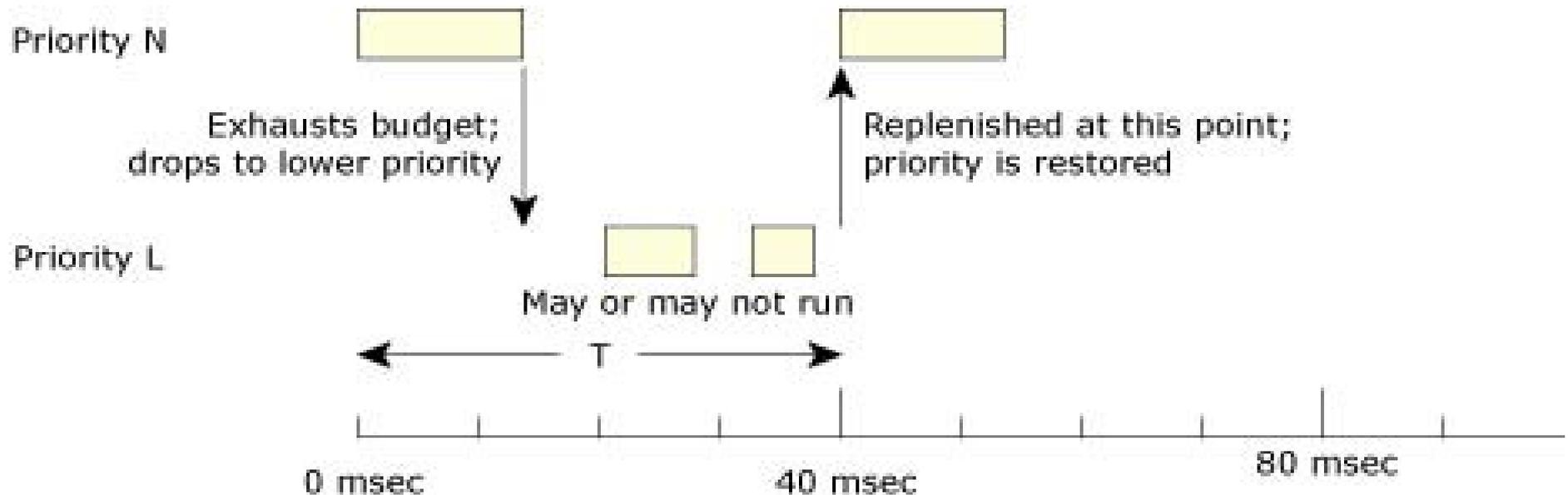
- Existem 2 níveis de prioridade
 - Normal e Baixo
- Toda thread é iniciada com um saldo
- A medida que a thread executa seu saldo é diminuído
- Quando a thread zera o saldo cai de prioridade
- O saldo é restaurado depois de um tempo T configurável

Sporadic



- Um processo executa
- Gasta parte de seu saldo e é bloqueado
- Volta a executar com saldo completo

Prioridades no Sporadic



Adaptative Partitioning Scheduler

- Processos rodam dentro de partições
- A cada partição é garantido uma porcentagem da CPU
- Prioridades são relacionadas aos processos de cada partição apenas
- Processos Críticos podem sobrepor as prioridades

Adaptative Partitioning Scheduler

- Execução de tempo real enquanto estiver dentro do saldo da partição
- Partições com mesma prioridade se sobrepõe
- O tempo livre de CPU é passado a partições mesmo que estas não sejam a prioridade no momento

eCos

- Criado em 1997 para ser um RTOS
 - open-source
 - altamente configurável
 - *royalty-free*
- Escalonamento de processos
 - Multilevel Queue
 - Bitmap
 - Lottery (experimental)

eCos

- Escalonador pode ser interrompido
- Pode-se também desativar o escalonador
- Chamada ao kernel para desativar escalonador
 - útil ao manipular interrupções
 - uma thread de modo usuário pode desativar o escalonador usando a chamada **cyg_scheduler_lock()**

eCos

- Multiple Queue
 - 32 filas de prioridade
 - O único algoritmo do eCos que suporta Symmetric Multi-Processing (Múltiplos Processadores)
 - Round Robin de múltiplas filas
 - Prioridade estática
 - Número de threads limitados apenas pela capacidade de memória

eCos

- Bitmap
 - É um algoritmo preemptivo
 - Funcionamento parecido com o Múltiplas Filas
 - Porém, cada Fila pode ter apenas um processo
 - Não existe o conceito de timeSlice
 - É bem mais simples e mais leve
 - Número máximo de processos limitado a 32

eCos

- Lottery
 - Cada thread mantém um número de loteria
 - A cada novo quantum um número aleatório é gerado
 - A thread sorteada é colocada para execução
 - Existem tickets de compensação para threads que bloqueiem antes do quantum
 - *Obs.: É um algoritmo usado apenas em testes do Kernel não deve ser usado em produção*

NetBSD

- Baseado no Unix de Berkeley
- Projeto irmão do FreeBSD
- É um SO pequeno cujo objetivo principal é ser altamente portátil e estável
- Slogan “Of course it runs NetBSD”
- Algoritmo de Escalonamento
 - Round Robin com filas múltiplas

NetBSD

- Round Robin de múltiplas filas
 - 32 filas de prioridade
 - threads são colocados nas filas cada um com sua prioridade da fila
 - cada fila é servida por um Round Robin comum
 - a medida que uma thread acumula tempo de CPU sua prioridade é alterada

MicroC/OSII

- Mantido pela Micrium Inc.
- Criado para ser um Kernel escrito em C com o mínimo de assembly
- Escalonamento baseado em prioridades
- Não poderá haver dois processos com a mesma prioridade
- O scheduler pode ser desativado

uClinux

- Algoritmo de Scheduling semelhante ao do Linux
- Seleção de processos Preemptivo
 - Round Robin Preemptivo
- Processos em modo de Kernel são Não-Preemptivos
 - Pode comprometer tempo de resposta para tempo real

Referências

- Tanenbaum, S. Andrew. **Modern Operating Systems**, 2 ed
- Cesati, M., Boveti, D. **Understanding Linux Kernel**
- Danko, A. **Adaptative Scheduler QNX**
- QNX Software, **QNX Migration Guide**
- **Manual NetBSD** acessado em:
<http://www.netbsd.org/docs/internals/en/index.html> 21 de março de 2008
- Wang, C. L., Yao B., Yang Y., Zhu Z., **A Survey of Embedded Operating System**
- Massa, Antonyh J., **Embedded Software Development with eCos**

Referências

- Hellström, D., **Manual: SnapGear Linux for LEON**
- **eCos Reference Manual** acessado em:
<http://ecos.sourceware.org/docs-1.3.1/ref/ecos-ref.2.html>
- Labrosse, Jean J. **MicroC/OSII The Real-Time Kernel**, CMP Books