

# Completely Fair Scheduler

Na perspectiva de Simulação de Eventos Discretos

Bruno E. O. Meneguele  
bmeneguele@gmail.com

Simulação de Eventos Discretos  
SIM0101

31 de Maio de 2019

# Escalonadores de Processos

## Tomada de Decisões

- ▶ Define as políticas (regras) para tomada de decisão;

# Escalonares em Sistemas Operacionais

- ▶ Define as políticas (regras) para tomada de decisão;
- ▶ Separação entre política e mecanismo (*dispatcher*);

# Escalonares em Sistemas Operacionais

- ▶ Define as políticas (regras) para tomada de decisão;
- ▶ Separação entre política e mecanismo (*dispatcher*);
- ▶ Diferentes abordagens para diferentes cargas de trabalho:
  - ▶ *Single* vs *Multi core*;
  - ▶ Usuário (*workstation*) vs Servidor

# Completely Fair Scheduler (CFS)

Escalonador do Linux 2.6.24+

# Alguns dos princípios do CFS

- ▶ Prioridade dinâmica em fila única;
- ▶ Eliminar situações de *starvation* com períodos fixos de execução;
- ▶ Considerar o quão “carregado” o sistema está;
- ▶ Considerar aplicações e não apenas *threads* (*grupo de tarefas*).

# Alguns dos princípios do CFS

- ▶ Prioridade dinâmica em fila única;
- ▶ Eliminar situações de *starvation* com períodos fixos de execução;
- ▶ Considerar o quão “carregado” o sistema está;
- ▶ Considerar aplicações e não apenas *threads* (*grupo de tarefas*).

Mas e as demais abordagens (i.e. FreeBSD ULE)?



# Alguns dos princípios do CFS

- ▶ Prioridade dinâmica em fila única;
- ▶ Eliminar situações de *starvation* com períodos fixos de execução;
- ▶ Considerar o quão “carregado” o sistema está;
- ▶ Considerar aplicações e não apenas *threads* (*grupo de tarefas*).

Mas e as demais abordagens (i.e. FreeBSD ULE)?

- ▶ Priorizar tarefas com alta prioridade e curta duração mesmo levando a situações de *starvation*;
- ▶ Filas múltiplas para “tipos” de tarefas: **interativa** vs **batch** e *idle*.

# Projeto de Simulação de Eventos Discretos do escalonador CFS

## Descrição do Modelo

**Servidores** representando **cores** e **fontes** gerando **tarefas** utilizando conceitos de sistemas *multi core*:

- ▶ Afinidade entre *cores* para grupo de tarefas;
- ▶ Controle e sincronização de acesso a recurso compartilhado: intra *core* e extra;
- ▶ ...

**Servidores** representando **cores** e **fontes** gerando **tarefas** utilizando conceitos de sistemas *multi core*:

- ▶ Afinidade entre *cores* para grupo de tarefas;
- ▶ Controle e sincronização de acesso a recurso compartilhado: intra *core* e extra;
- ▶ ...

Problemas:

- ▶ Complexidade da modelagem considerando-se Eventos Discretos;
- ▶ Tempo de entrega para disciplina.

**Servidor** representando **core** (processador) e **fonte** gerando **tarefas** utilizando conceitos de sistemas *single core*:

- ▶ Fila prioritária de execução única;
- ▶ Prioridade dinâmica para controle de *starvation*;
- ▶ Aplicações vs tarefas individuais: grupo de tarefas;
- ▶ Sistema preemptivo;
- ▶ Tarefas de tempo real **não** foram levadas em consideração;

Fonte: Aplicações que geram um número aleatório de tarefas;  
Servidores:

1. Processador;
2. Tratamento de Eventos.

Eventos:

1. Preempção (*slicetime* ou *quantum*): tempo variável com a prioridade da tarefa;
2. Dormir: requisição de uso de recurso externo, i.e. I/O, IRQ, ..., de ocorrência aleatória;
3. Acordar: tarefa volta à ativa após algum tempo aleatório.

# Variáveis do Modelo

## TAREFAS

<b>Variável</b>	<b>Distribuição</b>	<b>Justificativa</b>
Tempo de Criação	Uniforme	Uso contínuo do sistema
Quantidade	Normal	Aplicações geram mais de uma tarefa, mas de modo controlado
Prioridade	Exponencial	Considerando servidor, maior número de tarefas de baixa prioridade
Tempo de Execução	Uniforme	Tarefas semelhantes

## EVENTOS

Nome	Distribuição	Justificativa
Dormir	Exponencial	Servidores tendem a jogar o máximo de conteúdo em memória
Acordar	Normal	Depende do tipo de requisição foi feita: I/O, IRQ, ...



- ▶ *Runqueue*: Árvore Vermelha e Preta (*Red-Black Tree*);
- ▶ *Timeslice*: Baseado na prioridade;
- ▶ *vruntime*: Quantidade de tempo executado acumulado;
- ▶ Prioridade: diferença de *vruntime* é contabilizado;
- ▶ Criação de tarefas e eventos influenciam no *vruntime*

# Modelo

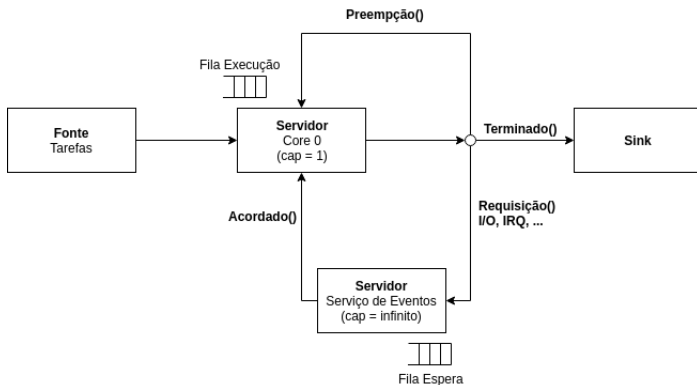


Figura 1: Modelo do escalonador CFS sob eventos discretos.

DÚVIDAS?

Muito Obrigado!

<https://github.com/bmeneguele/os-sched-des>

bmeneguele@gmail.com