

Sistemas Operacionais

Gestão de tarefas - escalonamento de tarefas

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Fevereiro de 2019

Conteúdo

- 1 Conceitos básicos
- 2 Escalonamento FCFS
- 3 Escalonamento RR
- 4 Escalonamentos SJF e SRTF
- 5 Escalonamento por prioridades fixas
- 6 Escalonamento por prioridades dinâmicas
- 7 Definição de prioridades

Escalonamento de tarefas

O escalonador define a ordem de execução das tarefas.

Tipos de tarefas:

De tempo real : exigem tempos de resposta precisos.

Interativas : recebem eventos externos e devem responder rapidamente.

Em lote : não têm requisitos temporais explícitos, executam sem intervenção do usuário.

Outra possibilidade de classificação:

CPU-bound : tarefas que usam intensivamente a CPU

IO-bound : tarefas que realizam mais entrada/saída que processamento.

Critérios de escalonamento

Métricas para avaliar diferentes escalonadores:

Tempo de vida (t_t): tempo entre a criação de uma tarefa e seu encerramento.

Tempo de espera (t_w): tempo perdido pela tarefa na fila de prontas.

Tempo de resposta (t_r): tempo entre a chegada de um evento ao sistema e a resposta a ele.

Justiça : distribuição adequada do processador entre as tarefas prontas.

Escalonamento preemptivo e cooperativo

Escalonamento cooperativo

a tarefa só perde o processador ao terminar, solicitar uma entrada/saída ou liberar explicitamente a CPU para voltar à fila de prontas (*syscall sched_yield*).

Escalonamento preemptivo

A cada interrupção, exceção ou chamada de sistema, o escalonador reavalia a fila de prontas e pode trocar a tarefa em execução.

Tarefas e métricas

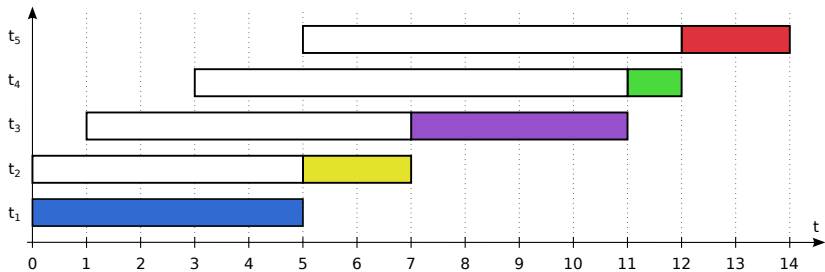
Tarefas a escalonar:

Tarefa	t_1	t_2	t_3	t_4	t_5
Ingresso	0	0	1	3	5
Duração	5	2	4	1	2
Prioridade	2	3	1	4	5

Métricas:

- Tempo médio de execução (T_t)
- Tempo médio de espera (T_w)

Escalonamento FCFS - *First Come, First Served*



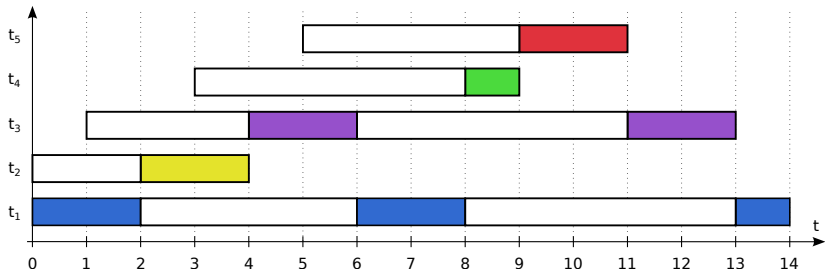
Escalonamento FCFS

$$\begin{aligned}
 T_t &= \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{5 + 7 + (11 - 1) + (12 - 3) + (14 - 5)}{5} \\
 &= \frac{5 + 7 + 10 + 9 + 9}{5} = \frac{40}{5} = 8,0s
 \end{aligned}$$

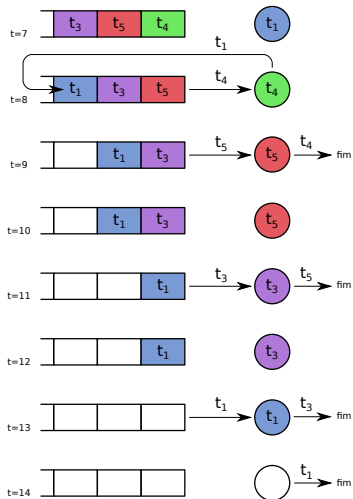
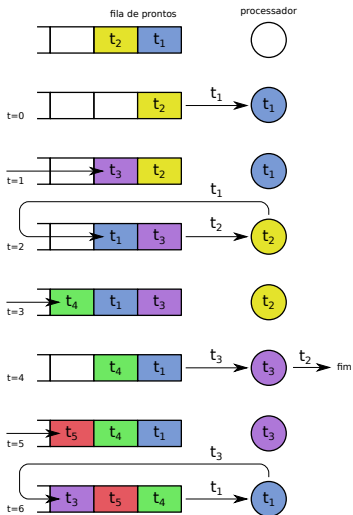
$$\begin{aligned}
 T_w &= \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{0 + 5 + (7 - 1) + (11 - 3) + (12 - 5)}{5} \\
 &= \frac{0 + 5 + 6 + 8 + 7}{5} = \frac{26}{5} = 5,2s
 \end{aligned}$$

Escalonamento RR - *Round-Robin*

Usa preempção por tempo (no exemplo, $t_q = 2$)



Escalonamento RR

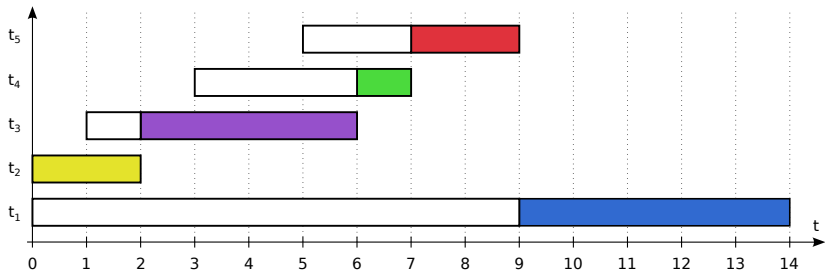


Escalonamento RR

$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{14 + 4 + 12 + 6 + 6}{5} = \frac{42}{5} = 8,4s$$

$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{9 + 2 + 8 + 5 + 4}{5} = \frac{28}{5} = 5,6s$$

Escalonamento SJF - *Shortest Job First*



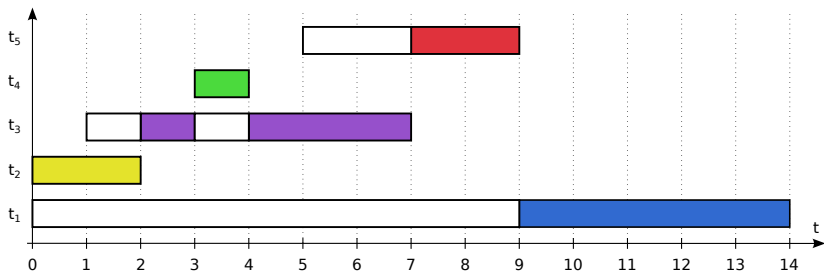
Problema: como definir/estimar a duração de uma tarefa?

Escalonamento SJF

$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{14 + 2 + 5 + 4 + 4}{5} = \frac{29}{5} = 5,8s$$

$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{9 + 0 + 1 + 3 + 2}{5} = \frac{15}{5} = 3,0s$$

Escal. SRTF - *Shortest Remaining Time First*

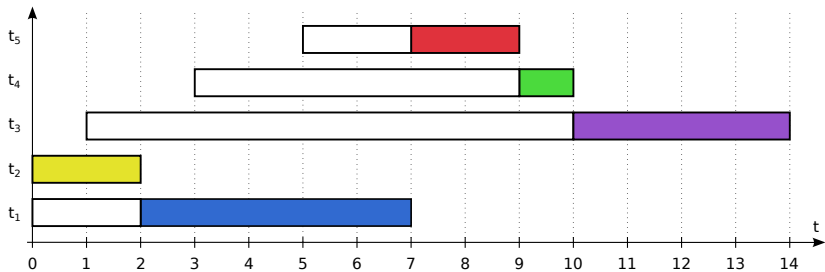


Escalonamento SRTF

$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{14 + 2 + 6 + 1 + 4}{5} = \frac{27}{5} = 5,4s$$

$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{9 + 0 + 2 + 0 + 2}{5} = \frac{13}{5} = 2,6s$$

Escalonamento PRIOc - cooperativo

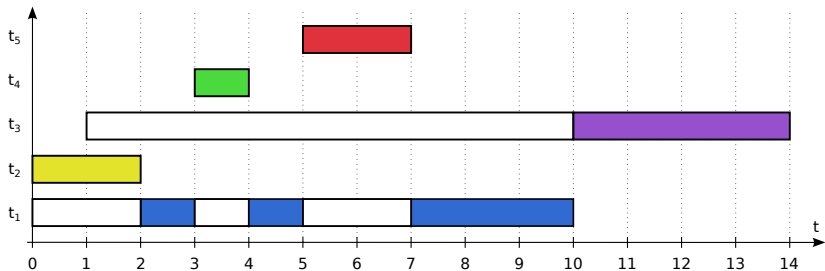


Escalonamento PRIOc

$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{7 + 2 + 13 + 7 + 4}{5} = \frac{33}{5} = 6,6s$$

$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{2 + 0 + 9 + 6 + 2}{5} = \frac{19}{5} = 3,8s$$

Escalonamento PRIOp - preemptivo



Escalonamento PRIOp

$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{10 + 2 + 13 + 1 + 2}{5} = \frac{28}{5} = 5,6s$$

$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{5 + 0 + 9 + 0 + 0}{5} = \frac{14}{5} = 2,8s$$

Escalonamento PRIOd - dinâmico

Problema:

- Tarefas de baixa prioridade têm pouco acesso à CPU
- Se houverem muitas tarefas, podem ficar paradas
- Fenômeno chamado “inanição” (*starvation*)

Solução:

- Aumentar gradativamente a prioridade de tarefas que não executam
- Ao executar, a tarefa volta à sua prioridade original
- Algoritmo de “envelhecimento” (*aging*)

Um algoritmo de envelhecimento

Definições: N : número de tarefas no sistema
 t_i : tarefa i , $1 \leq i \leq N$
 pe_i : prioridade estática da tarefa t_i
 pd_i : prioridade dinâmica da tarefa t_i

Quando uma tarefa nova t_{nova} ingressa no sistema:

$pe_{nova} \leftarrow$ prioridade fixa

$pd_{nova} \leftarrow pe_{nova}$

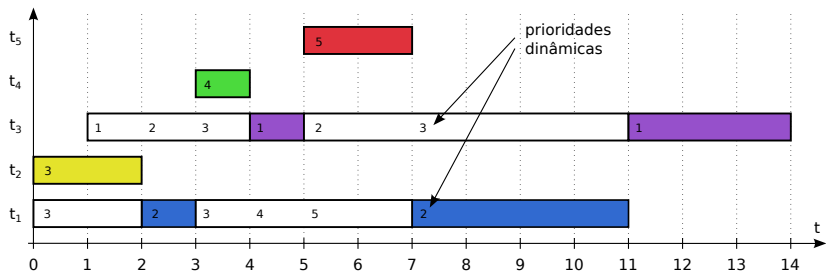
Para escolher t_{prox} , a próxima tarefa a executar:

escolher $t_{prox} \mid pd_{prox} = \max_{i=1}^N (pd_i)$

$\forall t_i \neq t_{prox} : pd_i \leftarrow pd_i + \alpha$

$pd_{prox} \leftarrow pe_{prox}$

Escalonamento PRIOd



Escalonamento PRIOd

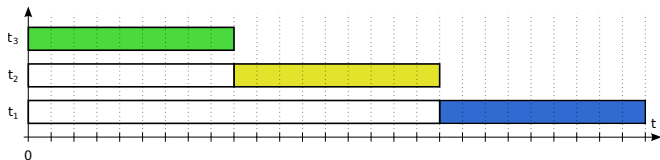
$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{13 + 2 + 13 + 1 + 2}{5} = \frac{31}{5} = 6,2s$$

$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{8 + 0 + 9 + 0 + 0}{5} = \frac{17}{5} = 3,4s$$

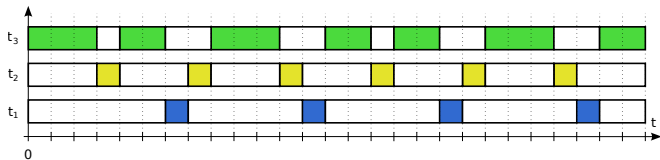
Efeito do envelhecimento

Round-Robin com $t_q = 1$ e $p(t_1) = 1, p(t_2) = 2, p(t_3) = 3$

Sem envelhecimento:



Com envelhecimento:



Quadro comparativo

Algoritmo	FCFS	RR	SJF	SRTF	PRIOc	PRIOp	PRIOd
Tempo médio T_t	8,0	8,4	5,8	5,4	6,6	5,6	6,2
Tempo médio T_w	5,2	5,6	3,0	2,6	3,8	2,8	3,4
Trocas de contexto	4	7	4	5	4	6	8
Tempo total	14	14	14	14	14	14	14

Definição de prioridades

Fatores externos:

- informações providas pelo usuário ou o administrador
 - classe do usuário (administrador, diretor, estagiário)
 - importância da tarefa em si (um detector de intrusão, etc)
- o escalonador não pode estimá-los sozinho
- Definem uma prioridade **estática**

Fatores internos:

- informações que podem ser obtidas pelo escalonador
- pode ser estimadas com base em dados internos
 - idade da tarefa
 - duração estimada
 - interatividade
- Permitem calcular uma prioridade **dinâmica**

Prioridades em sistemas Windows

Prioridades dos processos e threads entre 0 e 31:

- 24: *tempo-real*
- 13: *alta*
- 10: *acima do normal*
- 8: *normal*
- 6: *abaixo do normal*
- 4: *baixa ou ociosa*

A tarefa com janela ativa recebe mais prioridade (+1 ou +2).

Prioridades em Sistemas Linux

No Linux há duas escalas de prioridades:

- *Tarefas de tempo-real:*
 - Vai de 0 (mais importante) a 99 (menos importante)
 - Tem precedência sobre tarefas interativas
 - Somente o administrador pode criar tarefas de tempo-real
- *Tarefas interativas:*
 - Quase todas as tarefas dos usuários
 - Escala negativa de -20 (+ importante) a +19 (- importante)
 - Ajustável através dos comandos `nice` e `renice`.