

# Sistemas Operacionais

## Gestão de tarefas - escalonamento de tarefas

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Julho de 2020

# Conteúdo

- 1 Conceitos básicos
- 2 Escalonamentos FCFS e RR
- 3 Escalonamentos SJF e SRTF
- 4 Escalonamento por prioridades
- 5 Definição de prioridades
- 6 Escalonadores em sistemas reais

# Tipos de tarefas

Em relação ao comportamento temporal:

**De tempo real** : exigem tempos de resposta precisos.

**Interativas** : respondem rapidamente a eventos externos.

**Em lote** : não têm requisitos temporais explícitos.

Em relação ao uso de CPU:

**CPU-bound** : tarefas que usam intensivamente a CPU.

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

# Escalonamento de tarefas

## Escalonamento de CPU

Definir a **ordem de execução** das tarefas prontas.

Responsabilidade do **escalonador** de CPU.

Acionado pelo despachante durante as trocas de contexto.

# 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.

# Modos de escalonamento

## Escalonamento cooperativo

A tarefa só perde o processador ao terminar, solicitar uma entrada/saída ou liberar explicitamente a CPU (*syscall sched\_yield*).

Dica: o sistema só funciona se as tarefas cooperarem entre si.

## Escalonamento preemptivo

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

# Algoritmos de escalonamento

Tarefas a escalonar:

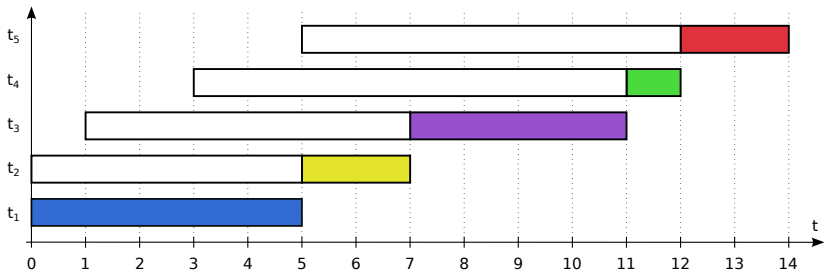
Tarefa	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
Ingresso (s)	0	0	1	3	5
Duração (s)	5	2	4	1	2
Prioridade	2	3	1	4	5

Métricas a considerar:

- Tempo médio de execução ( $T_t$ ) das tarefas
- Tempo médio de espera ( $T_w$ ) das tarefas

# Escalonamento FCFS

FCFS: *First Come, First Served*





## Escalonamento FCFS

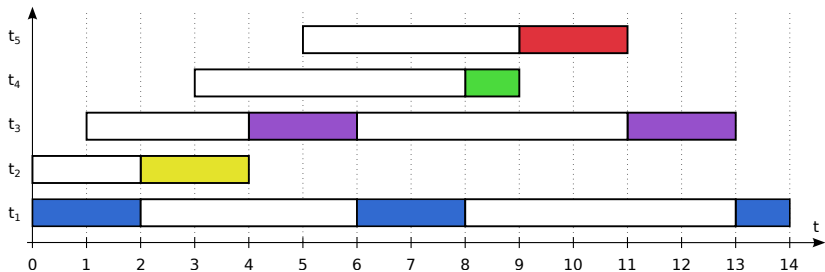
$$\begin{aligned}
 T_t &= \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{5 + 7 + (11 - 1) + (12 - 3) + (14 - 5)}{5} \\
 &= \frac{5 + 7 + 10 + 9 + 9}{5} = \frac{40}{5} = 8,0s
 \end{aligned}$$

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

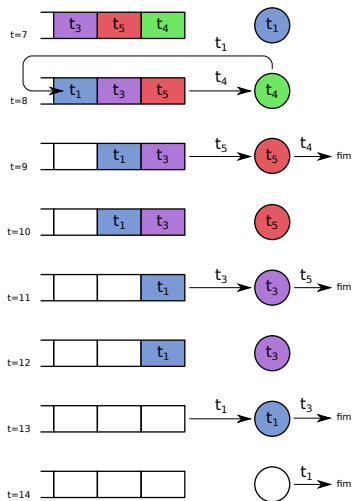
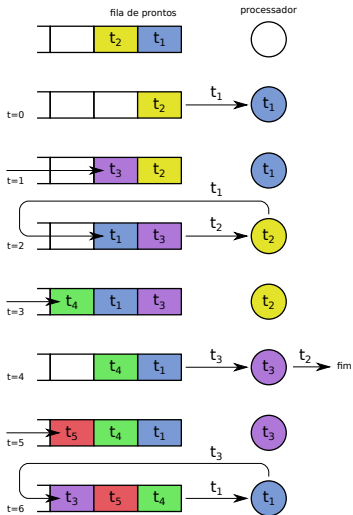
# Escalonamento RR

RR: *Round-Robin* (ou *rovezamento*)

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



# Escalonamento RR



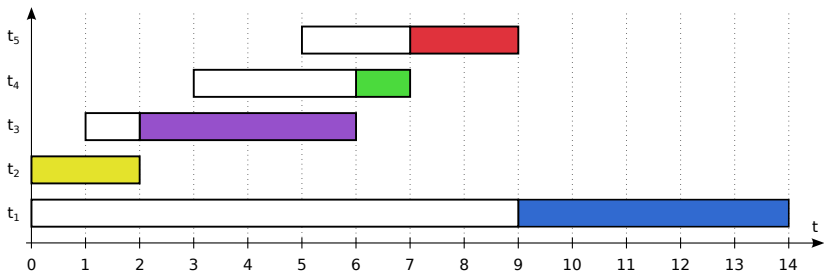
## Escalonamento RR

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

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

# Escalonamento SJF

*SJF: Shortest Job First*



## Escalonamento SJF

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

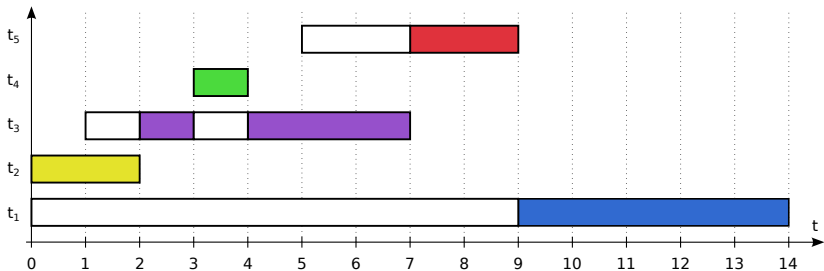
$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{9 + 0 + 1 + 3 + 2}{5} = \frac{15}{5} = 3,0s$$

Problema: **como prever** a duração de uma tarefa?

Pode ser usado em conjunto com *Round-Robin*.

# Escalonamento SRTF

SRTF: *Shortest Remaining Time First*



# Escalonamento SRTF

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

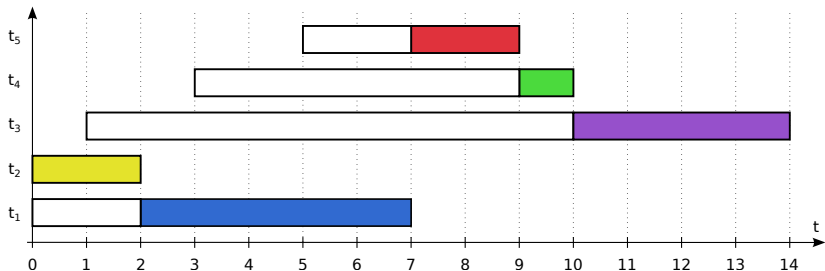
$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{9 + 0 + 2 + 0 + 2}{5} = \frac{13}{5} = 2,6s$$



# Escalonamento PRIOc

PRIOc: por prioridades, cooperativo

Prioridades:  $t_1: 2$   $t_2: 3$   $t_3: 1$   $t_4: 4$   $t_5: 5$



## Escalonamento PRIOc

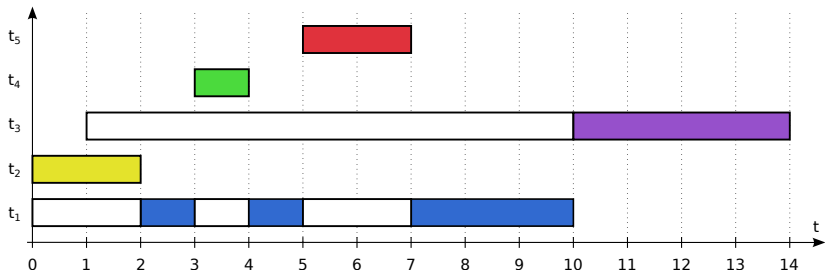
$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{7 + 2 + 13 + 7 + 4}{5} = \frac{33}{5} = 6,6s$$

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

# Escalonamento PRIOp

PRIOp: por prioridades, preemptivo

Prioridades:  $t_1: 2$   $t_2: 3$   $t_3: 1$   $t_4: 4$   $t_5: 5$



# Escalonamento PRIOp

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

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

# Prioridades dinâmicas

Problema das prioridades estáticas:

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

Solução: uso de **prioridades dinâmicas**

- Aumentar aos poucos a prioridade das tarefas paradas.
- 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  $t_{prox} \mid pd_{prox} = \max_{i=1}^N (pd_i)$

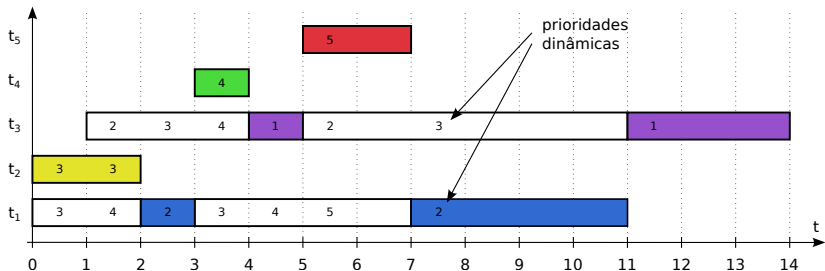
$pd_{prox} \leftarrow pe_{prox}$  (reinicia a prioridade)

$\forall t_i \neq t_{prox} : pd_i \leftarrow pd_i + \alpha$  (envelhece as demais)

# Escalonamento PRIOd

PRIOd: por prioridades, preemptivo e dinâmico

Prioridades:  $t_1: 2$   $t_2: 3$   $t_3: 1$   $t_4: 4$   $t_5: 5$



## Escalonamento PRIOd

$$T_t = \frac{t_t(t_1) + \dots + t_t(t_5)}{5} = \frac{11 + 2 + 13 + 1 + 2}{5} = \frac{29}{5} = 5,8s$$

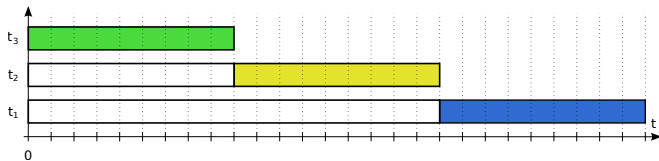
$$T_w = \frac{t_w(t_1) + \dots + t_w(t_5)}{5} = \frac{6 + 0 + 9 + 0 + 0}{5} = \frac{15}{5} = 3,0s$$



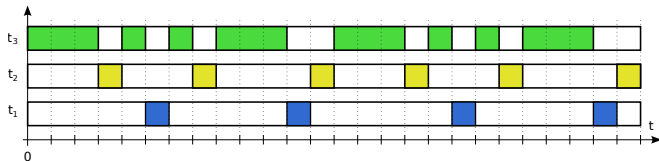
# Efeito da prioridade dinâmica

Round-Robin ( $t_q = 1$ ) e prioridades  $t_1 : 1$   $t_2 : 3$   $t_3 : 6$

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	5,8
Tempo médio $T_w$	5,2	5,6	3,0	2,6	3,8	2,8	3,0
Trocas de contexto	4	7	4	5	4	6	6
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**

# Definição de prioridades

## Fatores **internos**:

- Informações que podem ser obtidas pelo escalonador
- Pode ser estimadas com base em dados internos
  - Idade da tarefa
  - Duração estimada
  - Interatividade
  - Tempo em espera
- 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 1 (menos importante) a 99 (mais importante)
- Tem precedência sobre tarefas interativas
- Somente o administrador pode definir

- *Tarefas interativas:*

- Quase todas as tarefas dos usuários
- Escala negativa de -20 (+ importante) a +19 (- importante)
- Ajustável através dos comandos `nice` e `renice`

# Escalonador do Linux

Usa várias filas com políticas distintas:

**SCHED\_DEADLINE:** tarefas de tempo real com prazos, usa o algoritmo *Earliest Deadline First* (EDF).

**SCHED\_FIFO:** prioridades fixas preemptivo, sem *quantum*.

**SCHED\_RR:** SCHED\_FIFO + *Round-Robin*, *quantum* 10-200ms.

**SCHED\_NORMAL:** algoritmo CFS - *Completely Fair Scheduler* (*Round-Robin* c/ *quantum* variável, prioridades dinâmicas).

**SCHED\_BATCH:** tarefas *CPU-bound* de baixa prioridade.

**SCHED\_IDLE:** só recebe CPU se não houver tarefa ativa.